Revision, March 2004

A User's Guide to the Brave New World of Designing Simulation Experiments

Jack P.C. Kleijnen ● Susan M. Sanchez ● Thomas W. Lucas ● Thomas M. Cioppa

Department of Information Systems and Management /Center for Economic Research (CentER), Tilburg University (UvT), Postbox 90153, 5000 LE Tilburg, The Netherlands

Operations Research Department and the Graduate School of Business and Public Policy Naval Postgraduate School, Monterey, CA 93943-5219 USA

Operations Research Department Naval Postgraduate School, Monterey, CA 93943-5219 USA

U.S. Army Training and Doctrine Command Analysis Center, Naval Postgraduate School PO Box 8692, Monterey, CA 93943-0692 USA

kleijnen@uvt.nl ● ssanchez@nps.edu ● twlucas@nps.navy.edu ● tom.cioppa@trac.nps.navy.mil

---

Many simulation practitioners can get more from their analyses by using the statistical theory on design of experiments (DOE) developed specifically for exploring computer models. In this paper, we discuss a toolkit of designs for simulators with limited DOE expertise who want to select a design and an appropriate analysis for their computational experiments. Furthermore, we provide a research agenda listing problems in the design of simulation experiments—as opposed to real-world experiments—that require more investigation. We consider three types of practical problems: (1) developing a basic understanding of a particular simulation model or system; (2) finding robust decisions or policies as opposed to so-called optimal solutions; and (3) comparing the merits of various decisions or policies. Our discussion emphasizes aspects that are typical for simulation, such as the much larger number of factors than in real-world experiments and the sequential nature of the data collection. Because the same problem type may be addressed through different design types, we discuss quality attributes of designs, such as the ease of design construction, the flexibility for analysis, and efficiency considerations. Furthermore, the selection of the design type depends on the metamodel (response surface) that the analysts tentatively assume; for example, complicated metamodels require more simulation runs. We present several procedures for the validation of the metamodel estimated from a specific design, and provide a brief summary of a case study to illustrate several of our major themes. We conclude with a discussion of areas that merit further work in order to achieve the potential benefits—either via additional research or via incorporation into standard simulation or statistical software packages. A list with many references enables further study.

keywords: Simulation: Design of Experiments, Metamodels, Latin Hypercube, Sequential Bifurcation, Robust Design

---

1. Introduction

Design of experiments (DOE) has a rich history, with many theoretical developments and practical applications in a variety of fields. Success stories abound in agriculture, clinical trials, industrial product design, and many other areas. Yet, despite the impact DOE has had on other fields and the wealth of experimental designs that appear in the literature, we feel DOE is not used as widely or effectively in the practice of simulation as it should be. We suggest several possible explanations for this phenomenon.

One reason that DOE does not appear to be part of the standard "best practices" is that many simulation analysts have not been convinced of the benefits of DOE. Instead of using even a simple experimental design, many analysts end up making runs to measure performance for only a single system specification, or they choose to vary a handful of the many potential factors one-at-a-time. Their efforts are focused on building—rather than analyzing—the simulation model. DOE benefits can be cast in terms of achieving gains (e.g., improving average performance by using DOE instead of a trial-and-error approach to finding a good solution) or avoiding losses (e.g., obtaining an "optimal" result with respect to one specific environmental setting may lead to disastrous results when implemented). Unfortunately, many simulation practitioners seem unaware of the additional insights that can be gleaned by effective use of designs.

A second possible reason is that papers on DOE research are often found in specialty journals, making it difficult for simulation analysts to find out about the variety of methods available. Many papers make modifications that improve efficiency or guard against specific kinds of bias, whereas the bigger picture—namely, the setting for which this class of designs is most appropriate—may not be clear to an audience more familiar with simulation modeling issues than with statistical DOE.

The primary reason, in our opinion, is that most designs were originally developed for real-world experimentation and have been subsequently adapted for use in simulation studies, rather than developed specifically for simulation settings. Classic DOE textbooks (e.g., Box, Hunter, and Hunter 1978, Box and Draper 1987, Montgomery 1991, or Myers and Montgomery 2002) do not focus on the needs of simulation analysts, but instead on the practical constraints and implementation issues when conducting real-world experiments. Comprehensive simulation textbooks (Law and Kelton 2000, Banks et al. 2000) do cover a broad range of topics. Though they provide detailed lists of references, they demonstrate DOE by using it on a few simple test problems. These problems do not stretch the reader's mental framework as to the depth and breadth of insights that might be

achieved via other designs. So, studying classic DOE or general simulation textbooks familiarizes analysts with only a small subset of potential designs and applications; hence analysts are likely to force their problems to fit a particular design instead of identifying the design that best meets their needs.

Our goal is to bring together, in one place, (i) a discussion of the issues analysts *should* be aware of as they prepare to code, collect, and analyze output from a simulation model; and (ii) a guide for selecting appropriate designs. In particular, we contend that analysts must consider the following issues in order to come up with a truly effective analysis:

- the type of questions that they (or their clients) would like to answer;
- characteristics of their simulation setting;
- characteristics of, and constraints imposed on, the simulation data collection process;
- the need to convey the results effectively.

These issues seem straightforward, but we contend that there are some fundamental problems related to designing simulation experiments that are all-too-often overlooked. We discuss these more fully in the sections that follow, with a focus on the practical benefits that can be achieved through DOE. We believe that using a design suited to a particular application is much better than trial and error or limiting oneself to a simple, small design. Consequently, we should have no trouble convincing practitioners that DOE is a useful and necessary part of any analysis of complex simulation systems.

We do not intend this article to be a tutorial on the details for implementing specific designs, nor do we present a historical development of DOE and its application to simulation experiments. Instead, we try to provide an overview of the wide variety of situations that simulation analysts might face, the benefits and drawbacks of various designs in these contexts, and links to references for further details. Our overarching goal is to change the mindset of simulation analysts and researchers so they consider DOE to be an integral part of any simulation project.

This overview is based on our joint experience accumulated through contacts with many simulation users and researchers over the last few decades. Where we disagree with current practice and theory, we present both sides to further stimulate reflection and discussion. Despite the wide variety of designs that are available in the literature and—in some cases—statistical or simulation packages, we identify some situations where needs are still unmet. Our goal is to motivate further research to address these deficiencies.

In this paper we use concepts and terminology from simulation, statistics, and sensitivity analysis. Many readers may be familiar with simulation—and its DOE aspects—at the level of a textbook such as Law and Kelton (2000), who state that a "second course in simulation for graduate students" should cover their Chapter 12 on "Experimental design, sensitivity analysis, and optimization". For readers who may be familiar with only some of these ideas, we provide brief definitions and explanations as terms and concepts are introduced (and a few more details are available in the online companion to this paper). Those seeking a refresher or an overview of simulation experiments may find it beneficial to read Nakayama (2003) or Kelton and Barton (2003).

In Section 2 we describe how designing simulation experiments is different from designing experiments on real-world systems. Specifically, we address the types of questions that simulation analysts or clients should ask. We also describe a number of other characteristics of simulation settings that cannot easily be handled through more traditional methods, and provide examples to motivate the need for designs that cover a wide range of simulation settings. In Section 3 we discuss some characteristics of designs, including criteria that have been used to evaluate their effectiveness. In Section 4 we describe several classes of designs, and assess their strengths and weaknesses with respect to their appropriateness for various simulation settings and their design characteristics. In Section 5 we describe ways of checking the assumptions that were made when the experimental design was chosen. In Section 6 we present a small case study. In Section 7 we conclude with a discussion of areas that merit further work in order to achieve the potential benefits—either via additional research or via incorporation into standard simulation or statistical software packages. A list with many references enables further study.

2. Why is DOE for Simulation so Different?

First, we define some terminology. An *input* or a *parameter* in simulation is referred to as a *factor* in DOE. A factor can be either qualitative or quantitative. For example, consider a queueing system simulation. If queue discipline can be either LIFO (last in, first out) or FIFO (first in, first out), this is a qualitative factor. The number of servers is a discrete quantitative factor, while the rate for an exponential distribution used to model customer inter-arrival times is a continuous quantitative factor. In any case, each factor can be set to two or more values, called *factor levels*. Typically, factor levels are coded numerically for analysis purposes. A *scenario* or *design point* is obtained by specifying the complete combination of levels for all factors. We consider stochastic simulations, and hence *replicates* mean that different pseudo-random numbers (PRN) are used to simulate the same scenario. Unless otherwise specified, we will assume that these replicates use non-overlapping PRN streams, so that we have

independently identically distributed (IID) outputs across replicates—as most statistical methods assume. The output stream from a single replicate is a time series, which has auto-correlated observations.

Of course, the simulation is itself a model of some real-world (or prospective) system, process, or entity. We can view the simulation code as a *black box* that implicitly transforms input (such as factor level settings and PRN) into output. A *metamodel* is a model or approximation of this implicit Input/Output (I/O) function (a metamodel is also called a response surface, auxiliary model, emulator, etc.). When a simulation experiment is conducted and most or all of the factors are quantitative, one result is often the construction of a metamodel that characterizes the relationship between inputs and outputs in much simpler terms than the full simulation. One goal in choosing a particular design might be estimation of certain types of metamodels. A common metamodeling technique is that of polynomial regression. Here, it is assumed that the I/O relationship contains a deterministic (predictable) component, which is a polynomial function of the input factors, and a stochastic component that captures the (typically additive) error or randomness in the response. In a *first-order* or *main-effects* model, the deterministic component takes the form $f(X_1, X_2, ..., X_k) = \sum_{i=1}^{k} \hat{\beta}_i X_i$, where the $\hat{\beta}_i$ are estimated from the data. A *second-order* model could also include quadratic effects $\hat{\beta}_{i;i} X_i^2$ and two-way interactions $\hat{\beta}_{i:j} X_i X_j$. Higher-order models might also be defined, but are harder to interpret.

We emphasize the following chicken-and-egg problem. Once the design is specified and simulated, metamodel parameters can be estimated. On the other hand, the types of metamodels that the analyst desires to investigate should guide the selection of an appropriate design.

The field of DOE developed as a way to efficiently generate and analyze data from real-world experimentation. In simulation—with its advances in computing power—we are no longer bound by some of the constraints that characterize real-world experiments. This is both an opportunity and a challenge for those conducting simulation experiments. Indeed, it is an opportunity to gain much more insight into how systems behave, and so provide assistance and information to decision makers that might differ dramatically (in terms of its quantity and nature) from that obtainable using more traditional methods. It is a challenge because it may require a new mindset. Indeed, we argue that the way simulation experiments should be approached is now fundamentally different from the way that real-world experiments—involving, say, human subjects—should be approached.

To illustrate the difference between classic DOE and simulation DOE, we consider the classic bias-minimizing designs. For example, Donohue et al. (1993)—assuming a first-order metamodel, but at the same time allowing for possible bias caused by second-order effects—derive designs that minimize that bias. We argue that such designs are relevant in real-world experiments but not in simulation. In the former experiments, analysts must often select a design that is executed in "one shot" (say, one growing season in agriculture). In contrast, the data are collected sequentially in most simulation experiments, so analysts may start with a design for a first-order metamodel; then test (validate) the adequacy of that model; then augment their design to one that allows the estimation of second-order effects only if necessary (see, e.g., Sanchez et al. 1998, Kleijnen and Sargent 2000).

2.1 Asking Appropriate Questions

All simulation texts mention the importance of identifying the "right" problem before constructing a simulation and conducting the analysis. For example, Law and Kelton (2000) state that the first step in a simulation study is to formulate the problem and plan the study. As part of this step, they mention that the problem of interest should be stated by the project manager, and that the analysts should specify the overall study objectives, specific questions to be answered, performance measures that will be used to evaluate the efficacy of different system configurations, system configurations to be modeled, and the time and resources required for the study. They go on to say that experimental design, sensitivity analysis, and optimization deal with situations in which there is "…less structure in the goal of the simulation study: we may want to find out which of possibly many parameters and structural assumptions have the greatest effect on a performance measure, or which set of model specifications appear to lead to optimal performance." (Law and Kelton 2000 Chapter 12).

We recommend an even broader view, since we have found that the type of question people most often think about concerns an *a priori* single specific performance measure—typically a mean—which they then try to estimate or optimize. Instead, our starting point is three basic goals that simulation analysts and their clients may have:

- *developing a basic understanding* of a particular simulation model or system;
- *finding robust* decisions or policies;
- *comparing* the merits of various decisions or policies.

Developing a Basic Understanding

The first goal covers a wide range of questions. We use this phrase rather than "testing hypotheses about factor effects" for the following reason. At one extreme, we may be developing a simulation to gain insight into situations

where the underlying mechanisms are not well understood, and where real-world data are limited or even nonexistent. For example, when he was Chief Scientist of the Marine Corps, Dr. Alfred Brandstein posed the question "When and how should command and control be centralized or decentralized?" We do not know enough about the human mind to program a model for how decisions *are* made by an individual—let alone a group of people! Yet, ignoring these types of questions because they are "too hard" or "inappropriate for Operations Research" is unacceptable. Our profession's roots are in finding ways to address difficult, interdisciplinary problems.

One way that similar questions are explored is via the development of *agent-based* models that try to mimic and potentially explain the behavior of complex adaptive systems. For each agent (e.g., object or person), simple "rules" or "behaviors" are specified instead of building detailed prescriptive models to cover the interactions among all the agents. Applications of agent-based simulations have been developed to provide insights into the evolution of organisms, behavior of crowds in stadiums, swarming behavior of insects, food distribution, counter-terrorism activities, and more (see Horne and Leonardi 2001 for a discussion and examples of very simple agent-based models called distillations). For these types of simulations, DOE can be an integral part of the modeling development process. Indeed, we have found DOE useful in several ways. DOE can uncover detailed insight into the model's behavior, cause the modeling team to discuss in detail the implications of various model assumptions, help frame questions when the analysts may not know ahead of time what questions should be asked, challenge or confirm expectations about the direction and relative importance of factor effects, and even uncover problems in the program logic. These situations are typically not described in the open literature, particularly as they relate to problems in programming logic or modeling assumptions. To illustrate these benefits of DOE, we provide some anecdotal evidence from recent workshops and related research on the use of agent-based models for military decision-making (also see Sanchez and Lucas 2002, Horne and Johnson 2002, 2003, Cioppa et al. 2004).

Wan (2002) uncovered details about how a modeling platform behaved when simple terrain features were added. As part of the process of familiarizing himself with the modeling platform, he set up a skirmish within a corridor and used simple experimental designs to generate data for analysis. Wan expected that "barriers" would behave as walls—prohibiting movement and providing protection from fire—as in earlier agent-based combat distillations. Instead, he found instances where an enemy agent circled around the corridor and then exchanged fire with agents behind the front lines. Discussion with the software developer confirmed that these barriers prohibited movement but not fire, behaving as ditches rather than walls. This was a low-probability event in Wan's initial

scenario, and illustrates how DOE can uncover details about model behavior that might not be revealed without a broad-based investigation of factor effects.

Gill and Grieger (2003) ran several experiments to examine movement rules in time-step, agent-based modeling platforms. This led to a discussion of the implications of how various platforms implement an agent's movement when, e.g., the agent's propensity for moving toward a goal is "twice as high" as its propensity for avoiding enemy contact. This can impact the development of new scenarios by modeling efforts toward choosing appropriate weights for these propensities in different contexts. Without this investigation, scenario developers and analysts might all use the same term but have different internal views of its meaning.

When analysts may not know ahead of time what questions to ask, DOE can help frame interesting questions. For example, analysis of an agent-based model of a skirmish involving guerrilla forces (Red) attacking a conventional force (Blue) revealed that the most important determinants of both Red and Blue losses were factors associated with Red's stealth and mobility (Lucas et al. 2003). The scenario was initially set up to explore the merits of various Blue tactics, movement, and squad strength, but the findings suggest that Blue may gain more in terms of survivability and lethality by improving their ability to detect terrorists than by increasing their firepower.

Confirming prior expectations can be an important step in establishing face-validity for simulation models—as we illustrate in the case study in Section 6—but it is also informative when the simulation provides insights that do not agree with expectations. In early investigations of a model of maneuver through an urban environment where experimentation was limited to investigations of five factors at a time (Lucas et al. 2002), the factors believed by a small group of subject-matter experts to be the most important turned out to have no statistically significant impact on the results. Subsequent experiments provided interesting results, indicating that the commanders' propensities to maneuver toward friendly agents and away from enemy agents are critical and that losses are reduced when the commander balances his drive toward the goal (extraction point) with the avoidance of enemy contact. An interaction term indicated that a strong bond between the commander and subordinates could mitigate the negative impacts of so-called *friction* on the battlefield: even if the subordinate agents could not hear, comprehend, or otherwise act on their commander's orders, their losses were reduced if they stayed with him.

We have found that another major benefit of integrating DOE into the model development process is the ability to uncover problems in the program logic. For example, Barry and Forsyth (2003) describe anomalous results that, on further investigation, were due to a modeling artifact in the underlying movement component of an agent-based

model. A simple experiment revealed that movement speeds were not behaving as the analysts expected since increasing the factor setting from 0 to 1000 (corresponding to speeds of 0 to 10 squares per time step) did not result in monotonically increasing speeds. For example, if the movement setting was 110, then the agent would move two steps 10% of the time but remain in place 90% of the time, for an average speed of only 0.1 squares per time step. Identification of this problem led to its modification in subsequent versions of the software, yet Barry and Forsyth (2003) "wonder how often problems like this go undetected in more complex scenarios."

The benefits of using experimental design during the model development phase are likely to become even more in the future. If the ultimate decision-maker needs a rapid turnaround on the model development and analysis, this mandates using a modeling platform or reusing previously developed code or code modules in order to put together a model quickly. When code or modeling platforms are used or put together in new ways, it is not surprising to find that some details of the modeling logic are not well-documented or well-understood by the user.

In exploratory environments like those described above, it does not make sense to think about using the models to numerically estimate factor effects—we seek tendencies rather than values. At the other extreme, suppose we have a model that we are comfortable using for prediction. Then "understanding the system" may result from performing a detailed *sensitivity analysis* of a particular system configuration (i.e., examining the impact of small departures from this configuration). How should we proceed? Searching for effects by varying factors one at a time is an ineffective means of gaining understanding for all but the simplest systems. First, when using this approach it is impossible to identify any interaction effects (synergy or redundancy) between two or more factors. Second, even in the case when interaction effects are negligible so one-factor-at-a-time sampling provides valid insights into I/O relationships, this can be proven to be an inefficient way to estimate the factor effects. So, we assert that—from the outset—the analysts *must* explore factor effects concurrently in order to understand how their simulation model behaves when its factors are changed.

Between these two extremes of exploratory investigations and prediction generation are situations where the analysts wish to come up with a *shortlist of important factors* from the long initial list of potential factors. Depending on the context, this situation might lead to a more thorough investigation of this shortlist via additional simulation experiments, a decision to forego adding enhancements or greater detail to aspects of the model that were not found to be important, or the collection of (additional) real-world data in order to home in on appropriate values of (say) influential input distributions. Alternatively, simply identifying the most influential factors (and their

directional effects on performance) may suffice. It is also important to know which factors are "certainly" unimportant (at least over predetermined factor level ranges) so the users need not be bothered by details about these factors. Of course, the importance of factors depends on the *experimental domain* (or experimental frame, as Zeigler, Praehofer, and Kim 2000 call it). For example, oxygen supply is important for missions high in the sky and deep under water, but not on land at sea level. So the clients must supply information on the intended use of the simulation, including realistic ranges of the individual factors and limits on the *admissible scenarios*. This includes realistic combinations of factor values; for example, some factor values must add up to 100%.

Finding Robust Decisions or Policies

We discuss robust policies, rather than optimal policies, for a reason. It is certainly true that finding the *optimal* policy for a simulated system is a hot topic, and many methods have been proposed. These methods include heuristic search techniques that treat the simulation model as a black box—such as genetic algorithms, response surface methodology (RSM), simulated annealing, tabu search—and methods that analyze the simulation model to estimate gradients—such as perturbation analysis and score functions. The latter two techniques can be used in an optimization algorithm such as Stochastic Approximation. We refer to Fu (2002) and Spall (2003) for discussions of the current research and practice of optimization for simulation. Unfortunately, all these methods implicitly condition on a large number of events or environmental factors. In practice, the future environment is uncertain, so this so-called optimal policy cannot be achieved and may break down completely! Therefore, we wish to find a *robust* policy—that is, one that works well across a broad range of scenarios. Such policies have also been called "satisficing" (Simon 1981).

To illustrate this problem with classic optimization, consider using simulation to explore alternative layouts for a small factory. The project manager's decision factors relate to the type, number, position, and buffers associated with machines on the factory floor, as well as schemes for prioritizing or expediting orders. This is a prototypical problem often analyzed using a simulation optimization method—but the result of the "optimization" is conditioned on assumptions of specific (typically assumed independent) distributions for order arrivals, order sizes, machine uptimes, downtimes, and service times, and many more input variables. We argue that using the term "optimum" is problematic when the probability of all these assumptions holding in practice—even for a limited time—is zero. Suggesting different possible "optimal" layouts for several potential customer order patterns may be singularly unhelpful, since the decision-maker cannot control (or perhaps even accurately predict) future order patterns.

In contrast, a *robust* design approach treats all these assumptions as additional factors when running the experiment. These factors are considered noise factors—rather than decision factors—because they are unknown or uncontrollable in the real-world environment. A robust system or policy is one that works well across a *range* of noise conditions that might be experienced. Therefore, implementing a robust solution is much less likely to result in surprising (unanticipated) results. For example, Sanchez et al. (1996) used a multi-stage sequential approach to consider alternative factory layouts and order-dispatching rules in a job shop when the mix of demand for different products was unknown. They found that the "best" factory setup used neither the commonly used economic order quantity nor the just-in-time dispatching rule for batching orders. Their robust solution yielded a 34% smaller loss (in terms of expected deviation from the target total time in system) than would have been obtained if the goal had been to optimize the mean time in the system, and a 10%-22% improvement if the goal had been to optimize (minimize) the variance of the time in the system; the robust solution used no more (and usually fewer) total machines, indicating potential savings in capital expenditures. Another example is the study by Kleijnen and Gaury (2003) on robustness in production planning. They compare several solutions assuming a base scenario, and then find out which solution is least sensitive to changes in the environment.

We do not mean to imply that an optimization approach will necessarily yield a bad answer. An analyst can perform sensitivity analysis on any particular solution—either formally (e.g., applying DOE techniques or invoking mathematical arguments on the nature of the response surface) or informally (e.g., performing some trial-and-error investigations to determine whether small changes in the scenario lead to big changes in the output). If sensitivity analysis of the so-called optimal solution indicates that it still performs well (in an absolute sense) when realistic departures from these assumptions occur, then the optimization algorithm has identified a solution that is likely to perform well in practice. If changes in the environment (e.g., different patterns of customer orders) impact all potential solutions in the same manner, then the relative merit of particular policies does not change. If factor settings that are associated with good mean responses are also associated with good (i.e., low) response variances, then the solution found to be "optimal" in terms of mean performance will also be robust. A recent case study is Kleijnen et al. (2004), who derive a solution that minimizes both the expected value and the variance of the output of a supply chain simulation. That solution is controlled by the decision factors; the mean and variance are computed across several environmental scenarios.

Nonetheless, there are situations where optimizing and then performing sensitivity analysis can lead (and has led) to fundamentally different answers. For example, a military problem of interest these days is finding the "optimal" strategy for defending a high-value target (courthouse, church, or monument) against a single terrorist. If the analysts condition on the route the terrorist will take approaching the building, then forces will be concentrated along this path. Conversely, if the direction of approach is unknown, then an entirely different strategy (dispersing the protective forces) is much more effective. This *robust design philosophy* is inspired by Taguchi (1987), who pioneered an approach of using simple, orthogonal designs as a means of identifying robust product configurations for Toyota. The results improved the quality while lowering the cost of automobiles and component systems, because the chosen product designs performed well—despite variations in incoming raw material properties, the manufacturing process, and the customers' environments. This robust design approach is discussed for simulation experiments by Sanchez (2000). Metamodels can suggest scenarios (i.e., new combinations of factor levels) that have not yet been investigated—though the analyst should make confirmatory runs before applying the results.

Comparing Decisions or Policies

We avoid the phrase "making predictions about the performance of various decisions or policies". Comparisons may need to be made across a number of dimensions. Rather than formal statistical methods for testing particular factor effects or estimating a specific performance measure, our goal might be to provide the decision maker with detailed descriptive information. For example, we could present the means, variances, percentiles, and any unusual observations (see the box plots in Law and Kelton 2000) for the distribution functions of the estimators of several performance measures, for each of the systems of interest. These measures could then be reported, in conjunction with implementation costs and other considerations not included in the simulation model.

If at least some of the factors are quantitative, and if a performance measure can be clearly stated, then it is possible to construct metamodels of the performance that describe the I/O relationships in terms of functions of various factor levels. Here, rather than running an experiment in order to gain insight into how the performance is affected by all the factors, we may focus on a few of immediate interest to the decision maker.

If the analysts wish to compare a fixed small number of "statistical populations," representing policies or scenarios, *Ranking and Selection Procedures (R&S)*, *Multiple Comparison Procedures (MCP)*, and *Multiple Ranking Procedures (MRP)* can be used. There are two basic approaches: (i) how to select, with high probability, the system, decision, or policy that is, for practical purposes, the best of the group of potential systems; and (ii) how

to screen the potential systems/decisions/policies to obtain a (random-size) subset of "good" ones. Nelson and Goldsman (2001) provide a review of these procedures in simulation settings (see also Hsu 1996, Chick and Inoue 2001, and Goldsman et al. 2002). Many procedures have been developed specifically to address some of the characteristics of simulation experiments we will discuss in Section 3. Some assume that all populations are compared with each other, whereas others assume comparisons with a standard.

Summary

The three types of questions that we have posed differ from those problems others have suggested in the literature. Sacks et al. (1989) classify problems for simulation analysts as prediction, calibration, and optimization. Kleijnen (1998) distinguishes among sensitivity analysis (global, not local), optimization, and validation of simulation models. These two classifications are related to the ones we use—for example, global sensitivity analysis can be used as a way of gaining understanding about a problem—but there is not a one-to-one mapping. For certain classes of simulations, such as many military simulation models or hazardous waste disposal models, data are extremely limited or nonexistent. This means that calibrating, optimizing, or predicting may not be a meaningful goal.

In the best tradition of scientific discovery, we feel that simulation experiments can, nonetheless, have a role in supporting the *development* of insights (or theories) in these situations. For example, Helton and Marietta (2000) discuss how to assess the performance of a nuclear waste plant in New Mexico—in the next 10,000 years. Obviously, such a model is hard to validate—due to a dearth of data and changing conditions. Nevertheless, extensive sensitivity analyses convinced the Environmental Protection Agency (EPA) of the validity of this model, so it granted a permit to build and exploit this plant. Dewar et al. (1996) also discuss how one can credibly use models that cannot be validated in the simulation of future military conflicts—an area of almost unimaginable complexity. Despite the tremendous amount of uncertainty about potential future conflicts, decisions must be made (such as what equipment to purchase, how to organize units, and how to employ future forces) that will affect large sums of money and impact many lives. Since simulations of potential future force-on-force cannot be validated (Hodges 1991), these simulations are used to assist decision-makers in gaining insights into extremely complex systems and processes. For example, if the veracity of simulation cannot be ascertained, but it is known to favor one alternative over another, than *a fortiori* reasoning can be used to make a strong decision. Alternatively, if the outcomes of an unvalidated simulation are consistent with all the information that is available and seen as salient to the situation, then the simulation can be used to generate plausible outcomes. One can easily construct a case in

which an action would be avoided if a simulation suggests a potentially catastrophic outcome is plausible. More typically, high dimensional explorations of unvalidated models are used to help devise new ideas (i.e., tools for brainstorming) or to trace the consequences of assumptions over a variety of conditions.

The situations above contrast sharply with many of the simulation experiments that appear in the literature. These published accounts often presume that a thoroughly validated and verified simulation model exists, and the decision makers have very specific questions (for example) about the impact on a particular performance measure that results from changing a small number of factors to specified (new) values. The users might hypothesize the nature and strength of a particular factor effect, and the analysts' charge is to run the simulation model and collect I/O data in order to test this hypothesis.

2.2 The Simulation Setting

In this section, we describe some of the characteristics of simulation settings that call for *non-traditional designs* as part of the analyst's toolkit. To motivate our discussion, we include practical examples that we have recently worked on. These examples are drawn from industrial and military applications.

Number of Potential Factors

In real-world experiments, only a small number of factors are typically varied. Indeed, it is impractical or impossible to attempt to control more than, say, ten factors; many published experiments deal with fewer than five. Academic simulations, such as single-server queueing models, are also severely limited in terms of the number of potential input factors. Both application domains obviate the need for a larger set of designs in the analyst's toolkit.

For realistic simulations the list of potential factors is typically very long. For example, the MANA software platform was developed to facilitate construction of simple agent-based models (Lauren and Stephen 2002). The agents' rules for movement are a function of a "personality" or propensity to move. These rules are based on ten possible goals (toward/away from a location, a friend, an enemy, a road, etc.). The personalities can be in one of ten states and change on the occurrence of a trigger event (such as detecting an enemy or being shot at). In all, over 20 factors could be modified for each agent for each of the ten personality states, so we are dealing not with ten factors, but with thousands of factors—and this is considered a "simple" modeling platform! (Later versions of MANA allow 49 personality states.)

Other examples abound. Bettonvil and Kleijnen (1997) describe an ecological case study involving 281 factors. Cioppa (2002) examines 22 factors in an investigation of peace-enforcement operations. Even simple

queueing systems can be viewed as having a few dozen factors—if the analysts look at arrival rates and distributions that change over time, service distributions, and correlations arising when service times decrease or servers are added as long lines of customers build up.

We emphasize that good computer programming avoids fixing the factors at specific numerical values within the code; instead, the computer reads factor values so that the program can be run for many combinations of values. Of course, the computer code should check whether these values are admissible; that is, do these combinations fall within the experimental domain? *Such a practice can automatically provide a list of potential factors*. Next, the users should confirm whether they indeed wish to experiment with all these factors or whether they wish to *a priori* fix some factors at nominal (or base) levels. This type of coding helps *unfreeze* the mindset of users who would otherwise be inclined to focus on only a few factors.

Choice of Performance Measures

Consider both the type and the number of performance measures. Some problems require only *relative* answers; that is, whether or not one policy is better than another. For example, in a study on the search for sea mines, the users wanted to know which tilt angle of the sonar gives better results; see Kleijnen (1995). Conversely, some problems require *absolute* answers. For example, in the same case study, the users wanted to know whether the probability of mine detection exceeds a certain threshold—before deciding whether to do a mine sweep at all.

Most procedures (e.g., R&S, MCP, MRP, and RSM) involve a single quantitative performance measure; the goal is to maximize or minimize the expected value of this measure. However, in many simulation applications, it is unrealistic to assume a single measure that characterizes "the" system performance. For example, textbook examples of simple queueing systems often discuss minimizing the average waiting time. In practice, alternatives include minimizing the proportion of customers that wait more than a specified length of time, maximizing the number that are served within a particular length of time, improving customer satisfaction by providing information about their projected wait time and allowing them to reschedule, minimizing the number of errors in processing customer transactions, and balancing workloads across servers. Another example is provided by the various performance measures in supply chain management; see Kleijnen and Smits (2003). Consequently, it is restrictive to use a DOE framework that suggests the appropriate goal of the study should be examining the expected value of a single performance measure.

Taguchi's robust design approach (Taguchi 1987) offers another alternative in case of multiple performance measures. If responses are converted to losses and appropriately scaled, then the analysts can construct models of overall expected loss. We prefer constructing separate metamodels for each performance characteristic, because it makes it easier to identify *why* certain scenarios have more or less desirable performance than others.

A few researchers use a mathematical programming framework to analyze multiple simulation outputs; that is, one output is minimized whereas the remaining outputs should satisfy prefixed constraints. For example, the inventory is minimized while the service percentage meets a pre-specified level. See Angün et al. (2002).

Response Surface Complexity

Assumptions about the metamodel's complexity are generally broken down into those regarding its deterministic and its stochastic components, respectively. These assumptions often drive the analysis. The standard assumptions in the DOE analysis are that the deterministic component can be fit by a polynomial model of the factor levels (perhaps after suitable transformations of the factors or responses) and that the stochastic component can be characterized as additive *white noise*. The latter assumption means that the residuals of the metamodel are normally IID. In practice, normality may be explained by the central limit theorem. Unfortunately, the IID assumption is violated when the noise has larger variances in subspaces of the experimental area, which is known as *variance heterogeneity* or *heteroscedasticity*. Such heterogeneity is pervasive in simulations. For example, in queueing problems the intrinsic noise increases dramatically as the traffic load approaches 100% (Cheng and Kleijnen 1999, Kleijnen, Cheng, and Melas 2000). Moreover, common random numbers (CRN) are often used for generating output from several simulation scenarios, since CRN can sharpen the comparison among systems. Unfortunately, CRN violate the independence assumption.

Good modeling practice means that the analyst should strive to find the simplest metamodel that captures the essential characteristics of the system (Occam's razor). Therefore, we need a suite of design tools: some appropriate for simple response surfaces, others for more complex systems. We remark that simpler metamodels are often easier to justify when only a small number of factors and performance measures are examined; yet interpreting the results may be problematic because the analyst may easily miss important system characteristics. In Section 4, we will describe how some designs allow assessment of the suitability of the estimated metamodel. In principle, we prefer classifying factors into the following four categories: (i) factors thought to be very important, (ii) factors that might

be important, (iii) factors that are thought to be unimportant but are sampled anyway, and (iv) factors that we are quite comfortable in ignoring. Designs that sample differently across these classifications make intuitive sense.

It is increasingly apparent that some systems exhibit highly nonlinear behavior. A system's response surfaces may be characterized by localized regions where the response differs sharply from the surrounding area (spikes). It may contain thresholds (large, smooth contours in the factor space) where the response is discontinuous, so if the threshold is crossed the response steps up or down. It may contain regions over which the response is chaotic, that is, extremely sensitive to tiny changes in the input factor settings so that the output appears impossible to predict. For example, Vinyard and Lucas (2002) made billions of runs and found that chaotic behavior was rampant across many performance measures in a simple deterministic model of combat. Adding a stochastic component often mitigated this behavior, but sometimes aggravated some measures of the nonmonotonicity in the performance measures. Designs that examine only a small number of scenarios are unable to reveal such behavior; instead, the analysts may believe they are facing a simulation model with a large stochastic component.

Steady State versus Terminating Simulations

Terminating simulations are those that run until a specific event has occurred (including the event of simulating a fixed amount of time). Examples include simulating a single day's operation of a retail establishment, or a model of a space satellite that ends when the satellite is destroyed or becomes non-functional. In contrast, steady-state simulations have no natural termination point, so they can keep generating data for their analysis. The simulation type has implications on the design and analysis. For terminating simulations, it may be necessary to censor results if we are simulating rare events; see Kleijnen, Vonk Noordegraaf, and Nielen (2001). Multiple performance measures may again come into play; for example, it may be important to know not just who wins the battle, but how long it takes to finish. For steady-state simulations, the initial conditions are often chosen for convenience rather than relevance, e.g., a simulation of a computer network may start with all servers and relay nodes operational and no demands on the system. Here, the initial portion of the simulation output (or *warm-up period*) must be discarded or it may bias the results. The length of the warm-up period affects the total time required for experimentation, but is typically determined from pilot experiments or by analyzing the entire output stream.

Inclusion of Simulation-Specific Factors

The analysts have control over many things during the course of a simulation study—in addition to the factor levels they manipulate and the performance measures they collect. This control includes the maximum run time for

terminating simulations. For steady-state simulations this control includes specifying the warm-up period and run length(s), as well as how (if at all) the time-series output is averaged or aggregated into batches. The choice of the number of batches and batch sizes is an important topic of research in itself (e.g., Schmeiser 1982, Steiger and Wilson 2002) and an implicit assumption in many simulation analysis techniques is that appropriate batch sizes and warm-up periods are used. Other simulation-specific factors that can be controlled include the use of common random number streams (CRN) to facilitate comparisons across alternatives—for example, by subjecting all factory layout alternatives to the same pattern of customer orders. Other variance reduction techniques (VRT), such as control variates and importance sampling, have been developed for simulation output (see Law and Kelton 2000). Unfortunately, not all designs can easily accommodate these VRT.

2.3 External Concerns and Constraints

We now discuss issues that often play a major role in the implementation of simulation experiments, though they are generally not discussed in the literature.

Sequential versus One-shot Data Collection

In real-world experiments, the basic mindset is often that data should be taken simultaneously—unless the design is specifically identified as a sequential design. When samples must be taken sequentially, the experiment is viewed as prone to validity problems. The analysts must therefore randomize the order of sampling to guard against time-related changes in the experimental environment (such as temperature, humidity, consumer confidence, and learning effects), and perform appropriate statistical tests to determine whether or not the results have been contaminated.

Most simulation experiments are implemented sequentially—even if they are not (formally) analyzed that way. If a small number of design points are explored, this implementation may involve the analysts manually changing factor levels. Alternatively—and less prone to data entry errors—an input file or series of input files can be generated automatically, once a particular design has been chosen. These files may be executed sequentially (and efficiently) in batch mode. Modifying simulations to run in parallel over different computers is possible, but not typical. For example, parallelization is being used effectively at the supercomputing clusters of the Maui High Performance Computing Center and the Mitre Corporation in Woodbridge, Virginia. In many cases, "parallelization" results from an analyst manually starting different runs (or sets of runs) on a few assorted computers to cut down on the overall time to complete the data collection. For example, Vonk Noordegraaf et al. (2002) use five PCs at 533 MHZ to finish their 64 scenarios—each scenario replicated twice—in two weeks. We

remark that freely available software, such as that used on literally thousands of PCs as part of the search for extraterrestrial intelligence (SETI), could be used to facilitate parallel data collection for simulation experiments, but this is not yet readily available for use in either the industrial or academic settings with which we are familiar.

Premature Stopping of the Experiment

Another issue arises whenever the simulation takes a non-trivial amount of time to run. The analysts may have to terminate their experiment *prematurely*—because the computer breaks down, the client gets impatient, etc. An example is provided by the sequential bifurcation method (further discussed below) for finding the important factors among the many potentially important factors. This method identifies the most important factor first; then the next most important factor; and so on. Consequently, when the analysts must stop their experiment prematurely, they have already found the most important factors (including the magnitudes of their first-order effects).

We have found that this premature stopping occurs in many defense simulation projects. It is then better that the analyst organize the list of scenarios in such a way that the output can provide useful information, even if that input is curtailed. For example, consider a simulation model where a single input factor (taking the value 1 or 2) defines two systems the decision-maker wishes to compare. Further suppose that each run takes one day of CPU time, the design specifies 30 replications of each system, and a single computer is available (so it will take two months to complete the experiment). If the analyst conducts all 30 runs for system 1 before beginning the runs for system 2, they will be unable to tell the decision-maker anything about the relative merits of the alternatives until the end of day 31. Alternatively, an alternating sequence of runs would allow preliminary comparisons as early as the end of day 2, and half the data on each system would be available by the end of day 30. According to DOE theory the scenarios could be run in any order, but the latter approach is clearly preferable if preliminary results might be requested or the experiment might be terminated early. This idea also applies when runs are conducted on multiple machines and/or there are multiple input factors, each with multiple levels—provided a long time is needed to complete the experiment. With this view, even nonsequential designs can be implemented sequentially in ways that are robust to early termination. Clearly, sequential or partially sequential designs have this characteristic: after one stage of sampling the analysts indicate which configuration(s) should be examined next. Also, some single-stage designs can be viewed as augmentations of simpler designs, so there is a natural way to separate the designs into two or more parts (see the resolution 4 designs in Section 4).

Data Collection Effort

The information revolution has improved our ability to run simulations quickly. Simulations that used to take months now take hours; those that used to take hours now take seconds. This change has caused some analysts to add more details *to* their simulation models: we believe it should spur us to ask more *from* our simulation models.

Within the current computing environment, the traditional concept of a fixed sampling budget is unnecessarily restrictive. The primary indication of the data collection effort is likely to be the total time required to select a design, implement the design, and make the simulation runs. We have already described how parallel processing can reduce the total time. Even if a single computer is used, the time per run is not typically fixed. Different analysts might choose to use different run lengths and batch sizes. Run times might vary across scenarios because some tend to yield fewer events in steady-state simulations, or lead to early termination for non-steady-state simulations. The implementation effort is a function of both the time required to generate a design and that associated with setting the factor levels accordingly at the beginning of each run. Implementing a design may be very easy if software is available to generate coded factor levels, next convert them to original factor levels, and then generate input files so the simulation can be run in batch mode. Conversely, if the analysts must edit and recompile the code for each scenario—or make all changes manually through a graphical user interface—then the implementation time can surpass the time needed for making runs. We will discuss design choices in Section 4.

One way of describing this data collection effort has been to determine the time required to estimate the metamodel parameters to a certain level of precision. Unfortunately, it is difficult to use this time in making generic recommendations, since it depends on the underlying (heterogeneous) variability. In recent experience, we have dealt with simulations where run time varies from less than a second to half a day per scenario on a single processor.

A related issue is the trade-off between the number of design points and the number of replicates per design point. Suppose the total computer time is the same for the two options: one with many replicates per design point, and another with more design points and fewer replicates. The first option enables the explicit estimation of response variances that can vary across scenarios. If the primary goal of the study is *finding robust* systems or policies, then some replication at every design point is essential. If the goal is *understanding* the system, this may also include understanding the variance—again indicating that some replication is necessary. However, if the goal is that of *understanding* or *comparing* systems and a constant variance can be assumed, then this constant can be estimated using regression—provided no CRN are used and the metamodel is correctly specified. If classic ordinary

least squares regression is applied, it may be better to spend scarce computer time exploring more scenarios instead of getting more accurate estimators of the responses for fewer scenarios; i.e., the second option is preferred. Note that a single replicate yields an unbiased estimator of the response of a specific scenario. For example, consider a terminating simulation of a bank that closes at 5:00 p.m. The observed maximum queue length during a single day is an unbiased estimator of the true maximum. Of course, simulating more days provides a more accurate estimate based on the observed maximum averaged over all simulated days (though it does not change the fact that different scenarios may result in substantially different variances in the daily maximum queue length).

2.4 Conveying Results Effectively

The best experiment will come to naught if the results are not communicated properly to the decision-maker. We refer back to the three primary goals (developing a basic understanding, identifying robust solutions, and comparing systems). For the first goal, a good analogy is exploratory data analysis. Graphical tools that allow multi-dimensional visualization of the results may be much more helpful than equations or tables of numbers. Tools we have found useful include three-dimensional rotatable plots, contour plots, and trellis plots (Sanchez and Lucas 2002). Regression trees and Bayesian networks have also been effective ways of communicating which factors are most influential on the performance measures (Gentle 2002, Martinez and Martinez 2002). Yet, visualizing simulation results remains a challenge at this stage of simulation experimentation. Tufte (1990) is the seminal reference for excellence in graphical presentation; see also Meyer and Johnson (2001) for tools developed specifically for visually exploring large amounts of data from simulation experiments with multiple performance measures.

3. Criteria for Evaluating Designs

Once the simulation analysts know their situation, the question is: now what? Above we stated that there is no single prototypical situation (in terms of the type of question to be asked, or simulation characteristics) that analysts might face. In this light, it is not surprising that we cannot recommend a specific design. How, then, should analysts choose a design that is appropriate for their situation? While we do not have all the answers, we do attempt to provide some guidance.

In what follows, we use the term *design* to denote a matrix where the columns correspond to the input factors, the entries correspond to (possibly coded) levels for these factors, and each row represents a particular combination

of factor levels also called a *design point*. Readers wishing to see more detail on the construction and use of these designs can refer to this paper's online companion.

Others have listed desirable attributes for designs for experiments with real systems (see, e.g., Box and Draper 1987, Myers and Montgomery 2002). We describe some criteria that have been or might be used to evaluate designs in simulation settings, and discuss how they may (or may not) apply directly to the issues described earlier.

Number of Scenarios

In the literature, a major design attribute is the number of scenarios required to enable estimation of metamodel parameters. A design is called *saturated* if its number of factor combinations (say) n equals the number of metamodel parameters, q. For example, if the metamodel is a first-order polynomial in k factors, then q equals $k + 1$ (where 1 refers to the grand or overall mean, often denoted by β_0), so a saturated design means $n = k + 1$. Actually, there are several saturated designs for a given metamodel type. For the first-order polynomial in k factors, one saturated design changes one factor at a time, whereas another design is a fractional factorial (see Section 4 or Box, Hunter, and Hunter 1978). To choose among these different designs, we consider the following quality attributes.

Orthogonality

A design is said to be orthogonal if the columns of the design matrix are orthogonal. Orthogonality has long been a desirable criterion for evaluating designs. It simplifies computations. Since the input factors are uncorrelated, it is easier to determine whether or not to include them in a metamodel (e.g., using regression) and to separate their contributions to the overall fit of the metamodel. This in turn simplifies the interpretation of the results. Unfortunately, requiring orthogonality can have limitations as well. It may be that, in reality, some factor level combinations are not permissible. For example, in the M/M/1 queue the expected steady-state waiting time is infinite if the arrival rate exceeds the service rate. A complicated application (simulating part of the Rotterdam harbor) with exploding waiting times for the original orthogonal design appears in Kleijnen, van den Burg, and van der Ham (1979). In general, forcing the use of an orthogonal design may mean limiting many factors to narrower ranges, or figuring out a way to deal with unstable results at certain scenarios. Unfortunately, in complex models it may not be possible to know *a priori* which factor level combinations are problematic.

A design may be orthogonal in the *coded* factor values (such as -1 and +1), but not in the original factor values. Simulation analysts should be aware of possible scaling effects. Coding all the factor levels can facilitate identification of the most important factors (see, e.g., Box, Hunter and Hunter 1978 or Bettonvil and Kleijnen 1990).

Efficiency

The design determines the standard errors for the estimated metamodel parameters. The DOE literature uses several criteria (see Kleijnen 1987, p. 335). For example, *A-optimality* means that the sum of these standard errors is minimal. *D-optimality* considers the whole covariance matrix of the estimated parameters (not only the main diagonal); it means that the determinant of this matrix is minimal. *G-optimality* considers the mean squared error of the output predicted through the metamodel (also see Koehler and Owen 1996). Of course, these criteria require strong *a priori* assumptions on the metamodels to be fit to the data and the nature of the response (e.g., variance homogeneity). Consequently, they are of little value when there is substantial uncertainty *a priori* on the nature of the simulation's output.

The criteria above certainly can be—and have been—used to evaluate designs proposed for analyzing simulation experiments. Unfortunately, the classic DOE assumptions (polynomials with white noise) are usually violated in simulation. Moreover, focusing on minimizing the number of design points (or maximizing the efficiency for a fixed number of design points) may not be enough to insure "efficient" data collection, at least for steady-state simulations. In steady-state simulations, it does not make much sense to worry about using the most efficient design—if one does not also worry about using the smallest run length to achieve the desired goal. In short, efficiency is most critical when the runs are very time-consuming. Other criteria become more relevant when we are able to gather lots of data quickly.

Space-filling and Bias Protection

Conceptually, space-filling designs are those that sample not only at the edges of the hypercube that defines the experimental area, but also in the interior. A design with good space-filling properties means that the analysts do not need to make many assumptions about the nature of the response surface. At this point in time, space-filling designs provide the best way of exploring surfaces where we do not expect to have smooth metamodels. They are particularly useful for fitting nonparametric models, such as locally weighted regressions. These designs, especially Latin Hypercube Sampling (LHS), have indeed been applied when fitting Kriging models (see Section 4) and neural networks (see Alam, McNaught, and Ringrose 2003). The detection of thresholds is further discussed by Watson and Barnes (1995), who propose a sequential design procedure.

Space-filling designs also provide flexibility when estimating a large number of linear and nonlinear effects, as well as interactions, and so provide general bias protection when fitting metamodels of specific forms. Other

designs do not have good space-filling properties, but still protect against specific violations of model complexity assumptions; see the designs of resolution 3, 4, and 5 below. We also refer to the procedures in Sasena et al. (2002) and Kleijnen and Van Beers (2004a), who developed customize (but not space-filling) designs where sequentially selected scenarios were driven by the specific simulation application at hand.

Ability to Handle Constraints on Factor-level Combinations

In some situations (for example, chemical experiments) factor values must add up to 100%. The classic DOE literature presents *mixture* designs for these situations (Montgomery 2000). Many designs exist for exploring experimental regions (i.e., permissible combinations of design points) that are either hypercubes or spheres. In simulation experiments, restricting factor values to realistic combinations may complicate the design process dramatically. This is an area seriously in need of further research. Sanchez et al. (2001) propose elliptical designs, motivated by observational economic data. In many queueing situations, certain combinations of factor settings give unstable outputs (again see Kleijnen, van den Burg, and van Ham 1979, Sanchez et al. 2001). Until designs that can handle such situations are available, visual presentation of the results—and exploratory data analysis—may be the most appropriate ways of determining whether or not these situations exist.

Ease of Design Construction and Analysis

Designs should be easy to construct if they are to be used in practice. We will use this criterion in deciding which designs to recommend in Section 4. Nonetheless, some designs are useful even if they are difficult to generate, so we do not rule out the use of tabulated designs—particularly if they are incorporated into software packages so they can be easily retrieved by the user. The major statistical software packages include some experimental design generation methods. Ideally, design software should be readily available for many platforms. One example is "WebDOE," which helps users to design their experiments with deterministic simulation models—offering a library of classical designs through an easy-to-use Web interface (see http://www.webdoe.cc/).

The *analysis* is also easy if computer software is available for many platforms. Regression software is abundant, so the most common analysis tool is readily available and need not be discussed further. Newer surface-fitting methods are also available, including Kriging, neural nets, radial basis functions, splines, support vector regression, and wavelets; see Clarke et al. (2003) and Antioniadis and Pham (1998). Note that these are metamodel construction methods that can be applied to data collected using a variety of experimental designs, and may do a better job of fitting certain complex response surfaces. Because we have some experience with Kriging, which has

established a track record in deterministic simulation—and this metamodel is not yet much applied in random simulation—we briefly explain the basic approach (also see Van Beers and Kleijnen 2004).

Kriging is named after the South African mining engineer D.G. Krige, who developed his technique while searching for gold. It is an interpolation method that predicts unknown values of a random function or random process; see Cressie (1993)'s classic Kriging textbook or the excellent text by Santner et al. (2003). More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. These weights depend on the distances between the input for which the output is to be predicted and the inputs already simulated. Kriging assumes that *the closer the input scenarios are, the more positively correlated the outputs are*. This assumption is modeled through the correlogram or the related variogram. The optimal Kriging weights vary with the input value for which output is to be predicted, whereas linear regression uses the same estimated metamodel for all inputs to be predicted.

If the analysts are interested in the I/O behavior within a local area, then a low-order polynomial may be an adequate metamodel. However, for an experimental area that is global (not local), Kleijnen and Van Beers (2004b) demonstrate that a low-order polynomial gives very poor predictions compared with a Kriging metamodel. Giunta and Watson (1998) also compare Kriging with polynomial metamodels. Jin et al. (2000) compare Kriging with polynomial metamodels, splines, and neural nets. More recently, Van Beers and Kleijnen (2004b) apply Kriging to stochastic simulation; Jin et al. (2002) discuss the accuracy of Kriging and other metamodels under a sequential sampling approach.

Note that in *deterministic simulation*, Kriging has an important advantage over linear regression analysis: Kriging gives predicted values at observed input values that are exactly equal to the simulated output values. Deterministic simulations are used for Computer Aided Engineering (CAE) in the development of airplanes, automobiles, computer chips, computer monitors, etc.; see Sacks et al. (1989)'s pioneering article, and—for an update—see Simpson et al. (2001). Lophaven et al. (2002) have developed a Matlab toolbox for Kriging approximations to computer models, but the commercially-supported software products currently available (such as Kriging software in S+) are intended for real-world data, and so limited to three dimensions.

In theory, if a design is used to generate multiple outputs they will be accounted for in the analysis. For example, *multivariate regression analysis* may be applied. Each output is usually analyzed individually in practice.

For linear regression analysis, Khuri (1996) proves that this suffices if all outputs are generated by the same design. The same design is indeed used when running the simulation and observing multiple outputs.

4. Design Toolkit: What Works and When

Now that we have identified several characteristics of simulation settings and designs, it is time to match them together. Consider Figure 1, in which we chart some designs according to two dimensions that together describe the simulation setting. The horizontal axis represents a continuum from simple to complex response surfaces. Since the metamodel complexity depends on both the deterministic and stochastic components, there is not a unique mapping. We list some of the assumptions along the axis to inform the users about the types of metamodels that can be fit. The vertical axis loosely represents the number of factors. So, the lower left (near the origin of the figure) represents very simple response surfaces with only a handful of factors—that is, the traditional DOE setting with Plackett-Burman designs developed in the 1940s, etc. The upper right-hand corner represents very complex response surfaces with many factors. We do not present a comprehensive list of all available designs, but rather describe those that seem most promising and are either readily available or fairly easy to generate.

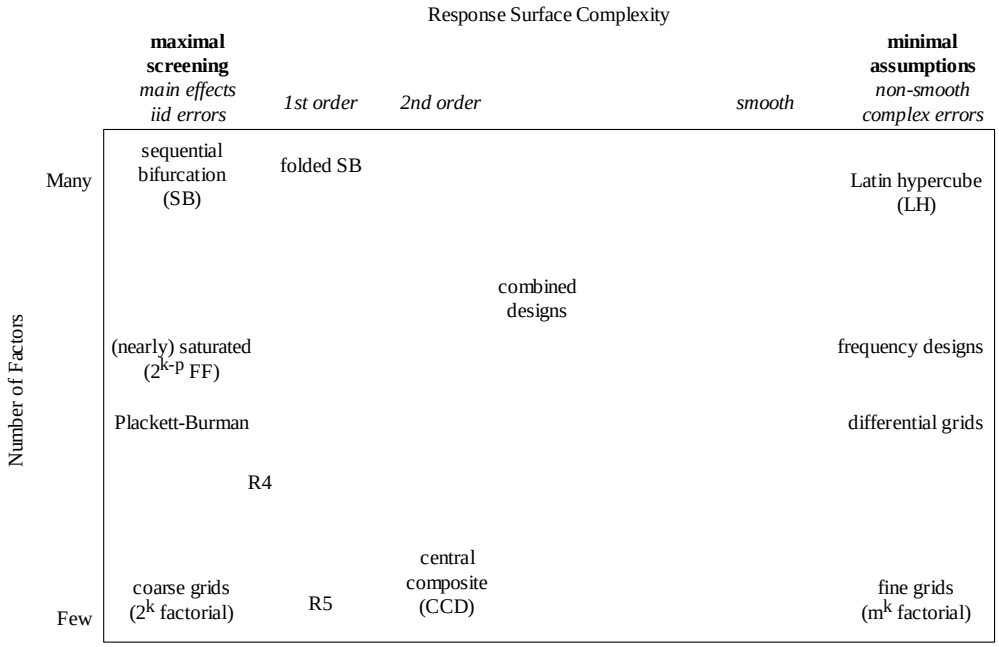

Figure 1: Recommended Designs According to the Number of Factors and System Complexity Assumptions

Recall that we hoped to change the mindset of those who might otherwise begin experimentation by focusing on a small number of factors. Therefore, we advocate using designs displayed near the top of this figure. In this

way, the analysts can look broadly across the factors in the simulation study. The analysts can start—from the left-hand side of the figure—making some simplifying assumptions, which will tend to reduce the initial data collection effort. (Of course, whenever assumptions are introduced, their validity should be checked later on.) Alternatively, employing CRN or other VRT can make certain procedures more efficient, and perhaps allow the analyst to handle more factors—making fewer assumptions—for a given computational effort. In our experience, VRT other than CRN seldom gave dramatic efficiency gains—except for rare-event simulations—but others report they have found VRT quite effective in large-scale simulations.

If this initial experiment does not completely address the main goal, then the analysts can use their preliminary results to design new experiments (augmenting the current data) in order to focus on the factors or regions that appear most interesting. This focus may mean relaxing metamodel assumptions for the shortlist of factors selected after the initial experiment, while holding the remaining factors to only a few configurations; that is, move south-east in Figure 1.

We now provide brief descriptions of the designs in Figure 1 and their characteristics, along with references for further details. A few sample designs are given in the online companion to this paper (also see Kleijnen 2004).

Gridded or Factorial Designs

Factorial designs are easy to explain to someone unfamiliar with classic DOE. A popular type of factorial is a 2^k design, which examines each of k factors at one of two levels, and simulates all resulting combinations. Then it is possible to fit a metamodel including *all* interactions—not only between pairs of factors, but also among triplets, etc. (these models are linear in the factor effects, not the factor levels).

Considering more complex metamodels (i.e., moving to the right in Figure 1), the analysts may use finer grids. Three levels per factor result in 3^k designs; in general, m levels per factor result in m^k designs. When there are more than a few factors, the analysts may use different grids for different groups of factors—employing finer grids for those factors thought to be important. These finer grids enable us to either view nonlinearities in the response surface or test the linearity assumption. Unfortunately, the number of scenarios n grows exponentially when k increases, so factorial designs are notoriously inefficient when more than a handful of factors are involved. Nevertheless, these designs are an important tool since they are easy to generate, plot, and analyze. Hence, whenever individual run times are minimal, the benefit of detailed information about the nature of the response surface may easily outweigh the additional computation time relative to the more efficient designs we discuss next.

Resolution 3 (R3) and Resolution 4 (R4) Designs

A design's resolution determines the complexity of metamodels that can be fit, with higher resolution designs allowing more complex models. Specifically, "a design of resolution R is one in which no *p*-factor effect is confounded with any other effect containing less than R - *p* factors" (Box, Hunter and Hunter 1978, page 385). Two effects are confounded when they cannot be separately estimated. For metamodels with main effects only (i.e., first-order metamodels with no interaction terms), it can be proved that the most efficient designs are R3 designs—provided the white noise assumption holds. If $k + 1$ is a power of two, R3 designs are fractional factorial designs—denoted as 2^{k-p} designs where the total number of design points is 2^{k-p}. If $k + 1$ is not a power of two but is a multiple of four, then R3 designs are tabulated as Plackett-Burman designs. See any DOE textbook for details (e.g., Box, Hunter and Hunter 1978).

If *interactions* are assumed to be present, but the users are mainly interested in estimating first-order effects, then R4 designs are appropriate. These designs give unbiased estimators of main effects—even if two-factor interactions are present. They can be easily constructed through the *fold-over* procedure; i.e., after executing the R3 design, the analysts run the mirror design that replaces each high value in a specific factor's column by its low value, and vice versa. In other words, the analysts can proceed in two stages by first running an R3 design and then augmenting it to an R4 design. (See also the RSM designs in Donohue et al. 1993.)

Even if the white noise assumption does not hold, classic designs do enable the analysts to obtain unbiased estimators of the metamodel parameters—although not necessarily with minimum standard errors. If we account for the analysts' time and energy, then these designs seem acceptable. Clearly, R3 designs give smaller standard errors for the estimated first-order effects than the popular practice of *changing one factor at a time*: the former designs use all scenarios to estimate all effects, whereas the latter designs use only two scenarios per effect.

Resolution 5 (R5) Designs

If users are also interested in the individual two-factor interactions, then an R5 design is needed. Many 2^{k-p} designs of the R5 type are not saturated. Saturated designs include the designs of Rechtschaffner (1967), which are discussed by Kleijnen (1987, pp. 310-311) and applied by Kleijnen and Pala (1999). The R5 designs requires $O(k^2)$ factor combinations, so these designs are less attractive if individual runs are time-consuming. If an R4 design suggests that certain factors are unimportant, then computing requirements can be reduced by limiting the R5 design

to fewer factors. The 2^{k-p} designs can be looked up in tables (see, e.g., Box et al. 1978, Kleijnen 1974-1975, 1987 or Myers and Montgomery 2002), but they are relatively easy to construct and so can be automated.

Fractional factorial designs (including R3, R4, and R5 designs) meet classic optimality criteria such as D-optimality for specific metamodels. Other designs that satisfy these criteria are derived in *optimal design* theory, pioneered by Fedorov and Kiefer; see Pukelsheim (1993) and also Spall (2003). These "optimal" designs typically lack the simple geometric patterns of classic designs, and are too complicated for many practitioners.

Central Composite Designs (CCD)

A second-order metamodel includes purely quadratic effects in addition to main effects and two-factor interactions. This means that *nonmonotonic* response functions can be handled. Best known are CCD, with five values per factor. These values are coded as 0, ± 1, $\pm c$, with $c \neq 0,1$. It is possible to determine an optimal value of c if the white noise assumption holds. Since this assumption does not hold for most simulation experiments, we do not worry too much about the choice of c—except to suggest the analysts choose an intermediate value for better space-filling. Details on CCD can be found in any DOE textbook (e.g., Box, Hunter, and Hunter 1978, Box and Draper 1987, Montgomery 1991, or Myers and Montgomery 2002).

Actually, estimation of quadratic effects requires no more than three factor levels, so to save computer time the analysts may again use *saturated* designs, which implies $n = 1 + k + k(k-1)/2 + k =$ (namely, 1 overall mean, k main effects, $k(k-1)/2$ interactions, k purely quadratic effects). Kleijnen (1987, pp. 314-316) discusses several saturated design types, including so-called simplex designs and fractional 3^k designs. Kleijnen and Pala (1999) apply simple saturated designs. Also see Batmaz and Tunali (2003).

Sequential Bifurcation (SB)

In practice, there are situations with a large number of factors but only a small number of really important factors. In such cases, a main-effects model—possibly augmented with two-factor interactions—may suffice. Moreover, the users may be able to specify the sign (or direction) of each potential main effect. In these situations, the individual factors can be aggregated into groups such that individual main effects will not cancel out. *Group screening* can be very effective at identifying the important factors. A practical and efficient group screening procedure is SB. For example, in an ecological case study, 281 factors are screened after only 77 factor combinations are simulated—resulting in only fifteen important factors; see Bettonvil and Kleijnen (1997). If interactions may be important, SB still gives unbiased estimators of the main effects—provided the number of combinations is doubled (similar to the

fold-over principle for R3 and R4 designs discussed above). If allowed to run to completion, SB will keep subdividing factor groups unless the estimated aggregate effect for a group is either nonsignificant or negative, or it identifies individually significant factors. However, SB can be stopped at any stage, and still provide upper bounds for aggregated effects—as well as estimates of any individual effects already identified. This means SB is robust to premature termination of the experiment. Bettonvil and Kleijnen (1997) discuss SB for deterministic simulations. Cheng (1997) extends the method for stochastic simulations. Kleijnen et al. (2004) also discuss SB for random simulations, including a supply-chain case-study. Wan et al. (2003) propose a modification, called Controlled Sequential Bifurcation, and provide proof of its performance under heterogeneous variance assumptions.

Other screening techniques with less restrictive metamodels are discussed by Campolongo et al. (2000), Dean and Lewis (2004), Holcomb et al. (2000a, b), Lin (1995), and Trocine and Malone (2001). Their performance relative to SB needs further research.

Latin Hypercube Sampling (LHS)

For situations involving a relatively large number of factors, McKay et al. (1979) proposed LHS: let *n* still define the number of scenarios; define *n* levels per factor; for each scenario, sample the factor values without replacement (giving random permutations of factor levels). LHS is so straightforward that it is incorporated in popular add-on software (such as @Risk) for spreadsheet simulation; see Sugiyama and Chow (1997).

LHS designs have good space-filling properties—particularly if several LHS designs are appended—so they are efficient ways of exploring unknown, but potentially complicated response surfaces with many quantitative factors. For LHS in Kriging—which assumes smooth metamodels, possibly with many local hilltops—we refer the reader to Koehler and Owen (1996), Morris and Mitchell (1995), Simpson et al. (2001), Pacheco et al. (2003), or Santner et al. (2003).

There are numerous variants of basic LHS. Recently, by assuming a linear metamodel Ye (1998) developed an algorithm for orthogonal LHS. Cioppa (2002) extended the number of factors that can be examined in orthogonal LHS within a fixed number of runs. Moreover, he found that by giving up a small amount of orthogonality (allowing pairwise correlations between the design columns less than 0.03), the analysts can dramatically increase the space-filling property of these designs. His LHS designs are not easy to generate, but are tabulated and thus—we think—useful in situations where individual simulation runs are time-consuming so the total number of runs is quite limited.

Frequency-based Designs

For quantitative factors, a frequency-based approach makes each factor oscillate sinusoidally between its lowest and highest value—at a unique and carefully chosen frequency. If the simulation model is coded so that factors can be oscillated during the course of a simulation run (called the *signal run*), then comparisons can be made to the *noise run* where all factors are held at nominal levels. This approach has been advocated as a screening tool for identifying important metamodel terms; see Schruben and Cogliano (1987) and Sanchez and Buss (1987).

More recently, frequency-based designs have been used to *externally* set factor levels for scenarios. That is, factor levels remain constant during the course of the simulation run—but they change from run to run; see Lucas et al. (2002) or Sanchez and Wu (2003). These designs have reasonably good space-filling properties. Moreover, there is a natural gradation in the granularity of sampling; i.e., factors oscillated at low frequencies are sampled at many levels, whereas factors oscillated at high frequencies are sampled at fewer levels. This property may help the analysts design an experiment to be robust to early termination, for example, by choosing higher oscillation frequencies for those factors believed *a priori* to be most important to investigate. By carefully choosing the oscillation frequencies, it is possible to use the results to fit second- and third-order metamodels. The designs are relatively easy to construct and to implement (see Jacobson, Buss, and Schruben 1991, Morrice and Bardhan 1995, Saltelli, Tarantola, and Chan 1999, or Sanchez and Wu 2003).

Crossed and Combined Array Designs

Selecting designs for finding *robust* solutions falls naturally into the upper middle portion of Figure 1. While there may be a large number of factors, the analysts are interested in a metamodel that captures the impact of the *decision* factors only. So their metamodel—while it may be complex—does not require estimation of all factor and interaction effects. Actually, the *noise* factors enter into the metamodel via their impact on the *variability* of the response for a particular combination of decision factor levels. This clear division of factors suggests that the analysts sample the two sets differently—for example, by crossing a 3^k or a CCD for the decision factors with a lower resolution design for the noise factors. This crossing means that each combination of decision factor values is simulated for each environmental scenario, which is defined by the combination of values of the environmental factors. These environmental scenarios enable the estimation of the mean and variance of the simulation response per combination of decision factor values. An alternative to a crossed design is a combined (or combined array) design (Shoemaker et al. 1991, Myers et al. 1992). In a combined design, a single design matrix (such as a factorial)

is used with columns divided among parameters and noise factors. As Myers et al. (1992) suggest, this can lead to a great reduction in the data collection effort since the only interactions that need be estimated are those involving two decision factors. Sanchez et al. (1996) apply both crossed and combined designs to explore a job shop simulation model. In Section 6 we illustrate the use of a crossed design to identify robust decision-factor settings in a small case study; the design is provided in the online companion.

Many types of designs have been used in this context. Taguchi (1987) proposes a particular class of orthogonal designs, but these designs are intended for factory experiments and are limited to main-effects models, which we find too restrictive for simulation environments. Ramberg et al. (1991) use a sequential approach, beginning with a 2^{k-p} augmented with a center point for the decision factors, and recommend a saturated or nearly saturated factorial for the noise factors. Moeeni et al. (1996) use three levels (varied across runs) per decision factor and frequency-based oscillation (varied within a run) for 35 noise factors. Cabrera-Rios et al. (2002, p. 225) propose three levels per decision factor and two levels per environmental factor. If the number of decision factors is not too large, then the analysts may cross a CCD for the decision factors with LHS for the noise factors; see the case study in Kleijnen et al. (2003). If the number of decision factors is large, then orthogonal or nearly orthogonal LHS may be a good design for the decision factors. In short, these designs are easy to generate, and the two subdesigns can be chosen to achieve the characteristics (space-filling, orthogonality, efficiency) that are most pertinent to the problem at hand.

Crossed designs can be exploited not only in robustness studies distinguishing between decision and noise factors. Lucas et al. (1997) give an example of group screening within a fractional factorial design crossed with LHS. Lucas et al. (2002) discuss the benefits of combining multiple designs after classifying factors into several groups based on their anticipated impact. This allows analysts much more flexibility than simply putting each factor into (or leaving it out of) the experiment.

Summary

We have presented several design alternatives for simulation experiments involving either a few or many factors. If runs are extremely time-consuming, then the analysts can reduce the computational effort by making some assumptions about the nature of the response surface. These assumptions can be checked after the runs are completed, as we shall describe in Section 5. We contrast this approach to arbitrarily limiting the number of factors. Indeed, if the analysts change only a few factors while keeping all other factors constant, then the conclusions of the simulation study may be extremely limited.

We remark that we have not attempted to list all the designs that have been proposed for simulation experiments. For example, we have not placed any simulation optimization methods in Figure 1, although "optimization" can be viewed as a means of comparing systems under very specific conditions. Our goal was to suggest some designs that analysts can readily use.

5. Checking the Assumptions

Whichever design is used, sound practice means that the analysts check their assumptions. If the analysts selected a design from the right-hand side of Figure 1, then they made very few assumptions about the nature of the response surface. In the process of fitting a metamodel, the analysts determine what (if any) assumptions are reasonable. If they started in the upper left corner of Figure 1, then the experiment was likely used to screen the factors and identify a shortlist as the focus of further experimentation. If so, the analysts are likely to make fewer assumptions during the next stages of experimentation. If they started from the lower left (as traditional DOE does), then it may be essential to confirm that the resulting metamodel is sufficient—or to augment it appropriately.

One check has the *signs* of the estimated effects evaluated by experts on the real system being simulated. For example, does a decreased traffic rate (resulting from adding or training servers) indeed reduce the average waiting time? Another example is the case study by Kleijnen (1995) on a sonar simulation experiment, in which naval experts evaluated the signs of the metamodel effects; because all these signs were accepted, the underlying simulation model was considered to be "valid". In general, checking the signs may be particularly applicable when the goal of the simulation study is general understanding rather than prediction, as for the agent-based models discussed earlier. We remark that sometimes intuition is wrong and needs to be challenged. For example, Smith and Sanchez (2003) describe a forecasting project where the model of losses (incurred for certain groups of loans) had the "wrong" signs. Examination of the detailed files confirmed that their patterns differed from the vast majority of loans and revealed why, so that the model ended up providing new—and valid—insights to the experts. Another example is the ecological case study in which Bettonvil and Kleijnen (1997) employ SB: the resulting shortlist of factors included some that the ecological experts had not expected to have important effects.

Another check *compares* the metamodel predictions to the simulation outputs for one or more new scenarios (which might be selected through a small LHS design). If the results are close, the metamodel is considered acceptable (see any textbook on linear models such as Neter, Wasserman, and Kutner 1985; also Kleijnen, Feelders, and Cheng 1998). Kleijnen and Sargent (2000) discuss how to use output from initial simulation experiments to test

the metamodel constructed from other scenarios in subsequent experiments. They refer to this as validating metamodels—not to be confused with validating a simulation model.

The assumption of normal IID errors can be examined via residual analysis (if regression is used to fit the metamodels), or by taking additional replications at a few design points. Tunali and Batmaz (2000) investigated procedures for validating this and other assumptions for least-squares metamodel estimation.

Note that *higher-order interactions* are notoriously difficult to explain to the users; nevertheless, traditional DOE routinely estimates and tests these interactions. One solution *transforms* the original inputs or outputs of the simulation model. We give two generic examples. First, logarithmic transformations of inputs and outputs may help in queueing problems; see Irizarry, et al. (2003) and Kleijnen and Van Groenendaal (1992, pp. 159-162). Second, replacing two individual factors by their ratio may help in queueing where the arrival and the service rates are combined into the traffic rate; in combat models the relative strength may provide a better explanation than the individual absolute strengths of the two combatants. Unfortunately, when multiple performance measures are collected, it may be difficult or impossible to transform individual *factors* so that all response surfaces are simple. One alternative might be to transform certain *responses*. We have observed instances where a transformation serendipitously yields responses of direct interest to the decision-maker (such as the differences in, rather than magnitudes of, sensor ranges), while allowing the analyst to fit simpler models in the transformed spaces. Presenting back-transformed results to the decision-maker is another option. Note that alternative estimation methods exist for those interested in making accurate predictions in an untransformed space (e.g., the MLE-delta method of Irizarry et al. 2003). Yet, those working in the area state that more experimentation and analysis are required to yield definitive concussions about how to handle the analysis of transformation-based metamodels. We find this concern about back-transformation less relevant in the context of developing a basic understanding, identifying robust strategies, or comparing alternatives when a fully-validated model may not exist and there is no clear delineation of the "best" output measure(s).

Even with careful thought and planning, it is rare that the results from a single experiment are so comprehensive that the simulation model and its metamodel(s) need never be revisited. In practice, results from simulation experiments often need to be modified; i.e., expanded or thrown out to obtain more detailed information on the simulation performance for a smaller region of the factor combinations. These modifications are determined in large part by the expertise of the simulation analysts. This points out a need for semi-automatic methods for

suggesting design refinements, which can be tricky. For example, suppose the analysts have built a response surface model that accurately characterizes simulation performance over a particular region of the factor space. Over time, the external environment changes so that the combinations of factor levels initially studied are no longer of primary interest—so some additional experiments are conducted. The question then is: when is it appropriate to use a global metamodel (with data from all experiments) instead of focusing on several local metamodels (over more restricted ranges)? This question merits further research.

6. Case Study: Humanitarian Assistance (HA) Operations

Clearly, no single investigation will use all of the experimental designs described in Section 4—even though they represent only a subset of possible designs. For the purpose of illustrating a number of the points made in the paper, we now present a small case summary of an investigation of an agent-based model of humanitarian assistance (HA) operations in urban environments. HA operations are of interest to Marine Corps logisticians for several reasons. In an HA environment one often sees services such as transportation, distribution, medical attention, and engineering efforts rise to the top of the priority list. Logisticians may find themselves in the unique position of being the main effort with infantry providing security for their missions, and must be prepared to take the lead in drafting operational plans. The ability to quickly put together agent-based models and explore their behavior over a wide range of parameter settings might help support operations during a humanitarian crisis. Marine Expeditionary Units deploy with a full compliment of equipment, which allows them to rapidly respond to pleas for help, provide immediate life-saving services, and then transition to relief and sustainment operations.

The scenario and initial analysis summarized in Section 6.1 were developed by Wolf (2003). In Section 6.2, we expand the investigation to more fully illustrate the central points of this paper. Examples of the experimental design matrices are provided in this paper's online companion.

6.1. Initial experiment

Wolf (2003) implemented a humanitarian assistance operation in MANA (Lauren et al. 2001). A convoy with a Marine security escort follows a given route to the southern of two humanitarian assistance sites in an urban environment. The convoy enters from the northeast corner, traveling west, and then turns south toward its final destination. Initially, the northern civilians make their way to the northern HA site, while the southern civilians move towards the southern HA site. As the northern civilians sense the trucks passing by, they speed up and try to follow the trucks. A lone aggressor searches for the convoy, provides harassing fire, and then runs away. The

security element will return fire if it identifies the aggressor, while the convoy responds by speeding up and driving out of the area. Once it reaches the southern HA site, the convoy begins distributing food to the civilians. The simulation runs for a fixed time. Initial conditions differ across runs due to random initial placement of agents within a defined border.

Wolf (2003) had several *goals* for this investigation that are consistent with those stated in Section 2.1. First, he was interested in demonstrating the potential use of agent-based models to support logistics operations, i.e., whether it was (or was not) feasible to use a modeling platform to rapidly develop a scenario and find things of interest to the decision-maker (see also Wolf et al. 2003). This meant from the outset that efficient experimental designs were needed in order to take a broad look across many factors simultaneously. Wolf also wanted to see whether gaining a better *understanding* of the model's behavior might offer some general insights to those interested in using agent-based models for HA/DR operations. He was interested in determining whether or not a *robust strategy* existed for the convoy and security element; that is, were there choices the Marines could make that would improve their ability to feed people over a broad range of environments? If no robust strategy emerged, he intended to examine a few different environments to see if he could identify appropriate strategies for more limited environmental conditions.

Forty combinations of squad/state/parameter (or factors) were chosen for the exploration. Here the 'squad' was a group comprised of one of the five agent types—northern or southern civilians, convoy, convoy security, or aggressor—implemented in the model. The 'state' corresponded to one of the states defining the behaviors for that squad (e.g., the convoy could be in the 'default,' 'shot at' or 'arrived at final waypoint' state). The 'parameter' represented a factor dealing with sensor range, communications delay, movement speed or precision, etc. The list of factors was carefully examined, while checking the documentation for parameter ranges, specifying the minimum and maximum values that made sense for the scenario, and noting whether the squad/state/parameter seemed important to examine closely or whether a simple check of any impact would suffice. For example, one of the simulation parameters (for any squad/state combination) was *movement precision*, which (according to the manual) could vary between 0 and 1000. Here 0 represented perfect precision, and 1000 represented a pure random walk. Though we were not directly interested in movement precision *per se*, we felt it could make a difference in how effective the operations were, e.g., by impacting how well the northern agents were able to notice and follow the convoy. Some imprecision might be beneficial if it keeps agents from approaching the northern HA site too

methodically and so missing the convoy as it passes by. Too much imprecision might mean that even if the agents spot the convoy, they might eventually lose contact because they are unable to follow it closely.

Few assumptions were made about the nature of the response surface, so the experimental setting falls in the upper right-hand corner of the design space in Figure 1. Had the number of factors been larger some factors could have been grouped together for screening purposes. Appending 16 square Latin hypercubes allowed us to examine the impact of simultaneously changing the specified values of these 40 squad/state/parameter combinations. Our final experiment (with 50 replications at each of 640 design points) required 32,000 simulation runs, but even this relatively large number of runs was completed in 7.5 hours on a computing cluster. The output measures were the numbers of northern and southern civilians fed, whether or not the red agent was killed, and whether one of the convoy trucks was destroyed.

For exploratory purposes, we began by averaging the responses at each design point and graphically assessing the results. A histogram of the total number of civilians fed indicated that while most of the time at least half of all civilians were fed, sometimes the operation was very unsuccessful. Further investigation of those design points associated with fewer than 35 civilians fed, on average, revealed that the associated design points all had the northern civilian *movement precision* setting equal to zero—which should have corresponded to no uncertainty in agents' movement beyond that introduced by the surroundings. The behavior at this point was dramatically different than for other low values, in both its magnitude and direction; see Figure 2. We then noted that if movement precision was set to zero in the graphical user interface, it was overridden with a value of one. This apparently did not happen when the program was run in batch mode. We modified the lowest precision factor setting from zero to one for the 16 design points in question, reran the 50 replications for each of these 16 design points, and replaced the old output with the new output before continuing our analysis.

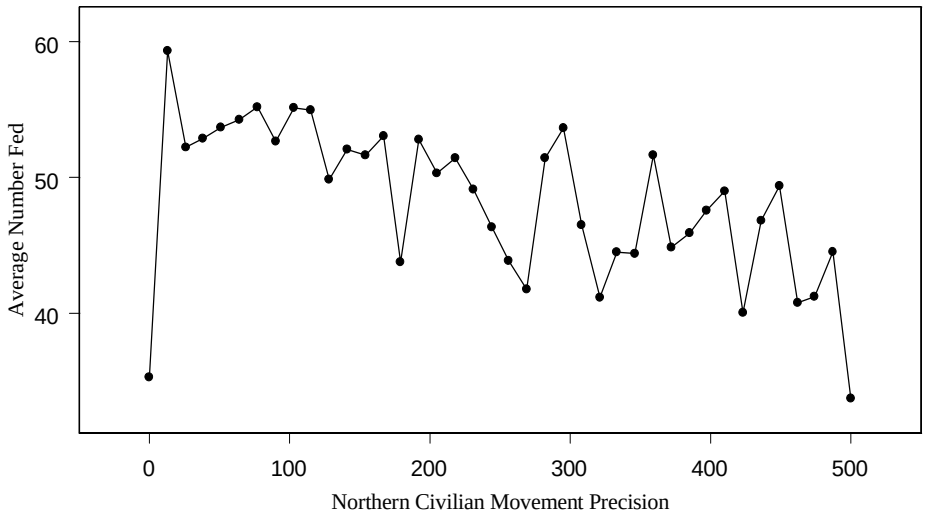

Figure 2. Average number of civilians fed vs. northern civilians' movement precision

Regression trees and regression models were also used to identify factors that seemed to play important roles in determining the response. We considered several types of models. First, we tried bounding the problem by fitting two "good" models: one involving only main effects, and another that used terms from a full second-order model. Here our criterion was to balance the model simplicity against the explanatory power. The associated R^2 values were 0.58 for a model with 11 main effects, and 0.63 for a model with a total of 20 terms (9 main effects, 3 quadratics, and 8 two-factor interactions).

We then addressed one of the initial questions of interest: how well can the Marines affect the success of the food distribution operation by specifying appropriate values for factors under their control? We found that these factors had very little impact on the performance (R^2=0.06) so no robust strategy emerged. However, after further discussion we decided that one other type of factor could be included as a Marine-specific factor. The sensor range associated with the northern civilian agents indicated how close they had to be to the convoy so they could perceive it (assuming it was within line of sight, not hidden by a building). This could also be viewed as a surrogate for the convoy broadcasting its presence while traveling. Adding the northern civilian sensor range to the Marines-only model improved it substantially (R^2=0.37, increase statistically significant with p-value < .001). We then considered models that explored only factors having to do with communication involving the civilian agents, and found that this explained even more. A model involving four main effects, one quadratic effect, and two interactions yielded R^2 = 0.45.

Since we were more interested in seeing how the model behaved than in making numerical predictions, we then examined contour plots for regions of interest. One such plot displays northern civilian sensor range versus convoy speed (an instance where interaction terms and quadratic effect were present in the regression metamodel). This plot seemed to indicate that when sensor range was sufficiently small, the convoy speed did not matter; see Figure 3. In this figure, the black dots indicate the design points, and lighter areas correspond to better performance (i.e., feeding a greater number of civilians). Note that the underlying relationships for our scenario may be somewhat smoother than illustrated by the contour plot because the settings of the 38 factors not shown differ across design points. This suggests that further data may be required if a subregion of interest appears to be highly non-linear.

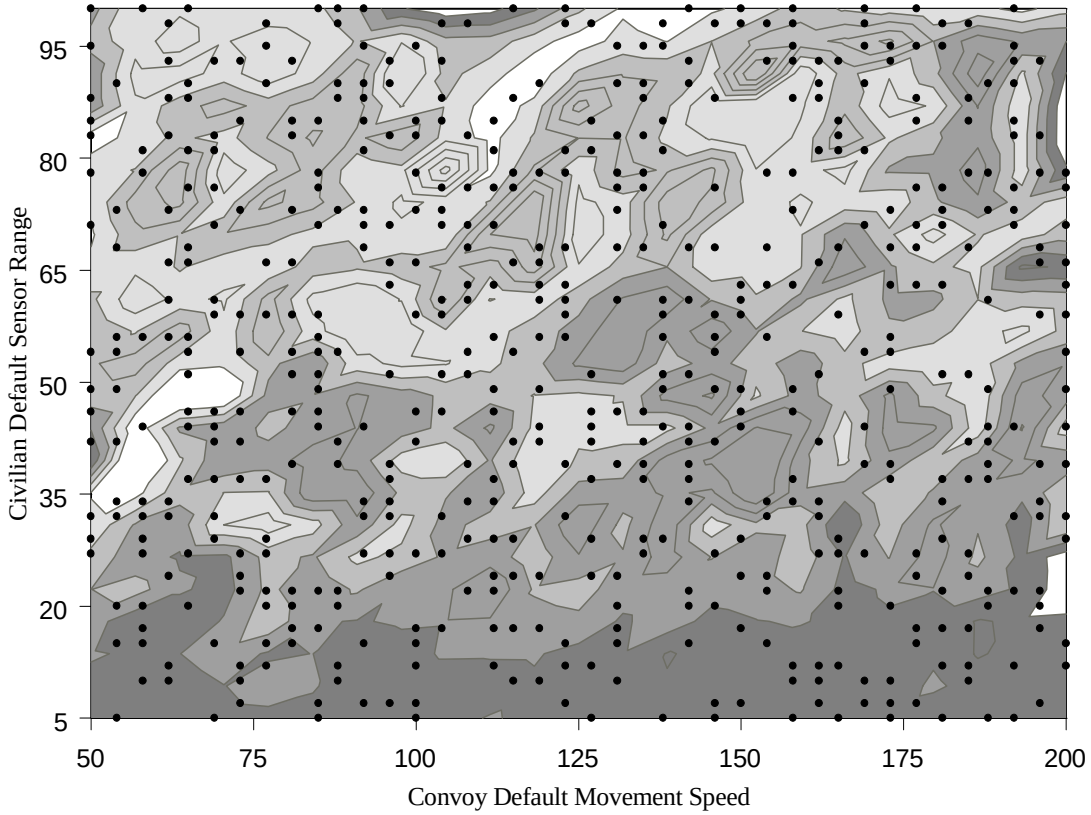

Figure 3. Contour plot: number fed as a function of northern civilian sensor range and convoy speed

6.2 Second Experiment

Based on results from the regression models and the contour plots, we decided to conduct a second experiment. Two of the original 40 factors were dropped (Wolf 2003), and the remaining 38 were divided into three groups: four

Marine-specific decision factors, seven environmental factors that had shown up in at least one of our models, and 27 environmental factors that had little apparent impact on HA operations in the initial experiment. We used an 11-run orthogonal LH design for the decision factors, and an 8-factor 33-run nearly-orthogonal LH design for the noise factors. Here we grouped all 27 not-so-interesting noise factors to form the eighth factor. We then crossed the two designs and ran 50 independent replications at each of the 561 design points for a total of 28,050 runs (details are provided in the online companion). Our main purpose was to facilitate comparisons in our search for robust solutions by ensuring orthogonality among the decision factors. The near-orthogonality of the noise factor design would make it easier to identify strategies that depend on noise factors (environmental conditions) deemed important in the initial phase of analysis. At the same time, embedding the other noise factors in the nearly orthogonal design would provide us with a check on our conclusion that these are less important. We kept the sample size roughly comparable to that of the first experiment so that results would be available within one working day.

We used squared-error losses to examine the robustness of the responses. This entails specifying a target τ that represents the "ideal" response value. In our case, a natural target is the total number of northern civilians seeking food—35 family units—although other values are possible. Mathematically, if μ_X and σ_X^2 denote the response mean and variance at a particular design point *x*, then the expected loss is $\sigma_X^2 + \left(\mu_X - \tau\right)^2$ (see Ramberg et al. 1991, or Sanchez et al. 1996). We computed the average losses for each of the 17 design points. The maximum average number fed corresponded to row 1 in our decision-factor matrix, but was still far from the highest possible value of 35. Rows 8 and 13 in our decision-factor matrix were within 8% and 11% of the most robust solution, respectively. The other design points were far from robust: their average losses were between 65% and 290% higher than that of the robust solution.

While the above analysis allows us to compare the robustness of specific design points, regression can be used to suggest alternatives that might perform even better. Accordingly, we fit a second-order model of the average number fed involving only the decision factors. $R^2 = .33$ for the full second-order model, and a model with three main effects and two quadratic effects worked essentially as well (R^2=0.328, all p-values < 0.05). We also fit models of ln(Var(number fed)) and settled on a model with two main effects and two quadratic effects ($R^2 = .80$, all p-values < 0.05). The results suggest, for example, that with a target value of 20, good tactics are to set the convoy default movement speed, northern civilian sensor ranges (in both the default and contact states) to their highest

levels. Note that the three most robust decision points correspond to these same three factors set to relatively high levels (average of 12.7, 14.7, and 13 for design points 1, 8, and 12, respectively). This complements the robust comparison, and suggests an additional alternative that could easily be checked with a set of confirmation runs.

Finally, since we cross an orthogonal design matrix with a nearly-orthogonal one, we can assess the impact of adding noise (environmental) factor terms to our regression model—without worrying about multicollinearity. Adding four of the noise factors and ten decision-by-noise interactions to the decision-factor model increases R^2 from 0.33 to 0.76 (all p-values < 0.05) with the decision-by-noise interactions accounting for about 0.08 of the improvement. Further examination of the signs associated with the noise factors and interactions indicate that setting all factors to their low levels is a favorable environment for the relief efforts, while setting all to their high levels is unfavorable. This could be a first step in adapting the convoy's tactics to suit the environment.

One general lesson learned is that the environment is critically important for the success of HA/DR operations. The relief efforts can be improved with better communication, both between the convoy and those civilians in need of assistance, and among the civilians. The possibility of combining agent-based models and infrastructure—that allows data to be generated quickly and systematically—makes this approach promising for short-turnaround decision-support of logistics operations.

7. Conclusions and Future Research

Our primary goal in writing this paper is to help change the *mindset* of simulation practitioners and researchers. Indeed, we believe that practitioners should view DOE as an integral part of any simulation study, while researchers should move beyond viewing the simulation setting merely as an application area for traditional DOE methods. We advocate thinking first about three potential goals of a simulation experiment, namely (i) understanding a system, (ii) finding robust solutions, or (iii) comparing two or more systems. We contend that these are often more appropriate goals than those typically used; namely, testing hypotheses about factor effects, seeking an optimal policy, or making predictions about performance. To illustrate our points, we describe examples from decades of combined experience. We also describe many characteristics of the simulation setting that call for nontraditional designs as part of the simulation analyst's toolkit. In particular, simulation experiments are often characterized by a large number of potential factors, complex response surfaces, time-varying correlated output streams, and multiple performance measures. Analysts also have the opportunity to control simulation-specific factors (such as run lengths, random number streams, and warm-up periods) that can be exploited for further design efficiencies.

Steady-state simulations offer the possibility for batching output and/or conducting very long runs that may not have useful analogs in real-world experiments.

Another change in mindset occurs when analysts begin thinking explicitly about sequential experimentation. This has two major implications. First, it means that a sequence of experiments may allow the analyst to gather insights efficiently. Second, even for one-shot experiments, it may be beneficial to sequence the simulation runs appropriately in order to allow for useful partial information as preliminary results become available or in case the experiment is halted prematurely. We argue that the data collection effort consists of the number and length of the simulation runs, as well as the effort required to generate the experimental designs and manage the runs. Emphasizing solely the former may unnecessarily limit the choice of experimental designs. A related idea is the benefit of coding the simulation model in a way to facilitate creating a list of potential factors and subsequently modifying their levels. At the same time, conveying the results effectively remains a challenge for high-dimensional response surfaces.

We discuss several criteria for evaluating designs, and provide guidance on selecting a design suitable for a particular context and using them appropriately. A small case-study of a humanitarian assistance operation illustrates several of our major points.

In this paper, we have listed many problems that require more investigation, resulting in a *research agenda for the design of simulation experiments*. For example, it is important to further investigate sequential design and analysis since most computer architectures simulate the scenarios and replicates one after the other. The issue of "robust" instead of "optimal" solutions requires further research. Further work on better matching types of metamodels (and appropriate designs for developing these metamodels) to the characteristics of the simulation setting will continue to be important to analysts and decision-makers. Screening designs deserve further investigation and application, particularly if they can be incorporated into other designs to reduce the large number of factors at the start of the investigation. Nonsmooth metamodels are needed to represent spikes, thresholds, and chaotic behavior; appropriate designs require more research and software. Multiple outputs might need special designs and analyses for alternative metamodels—such as Kriging and neural nets—and for evaluating or comparing systems. In addition, approaches that deal with constraints on factor level combinations and unstable system configurations are critical if we are to explore large regions of the factor space.

In addition to the research, better software is needed to provide support for appropriate design and analysis methods. While gains have been made in recent years—as in visualization software, Kriging, and data-mining tools—there is still much room for improvement. This means there are challenges that remain for simulation modelers, software developers, simulation consultants, and analysts. Modelers who use general-purpose software should incorporate sound programming techniques that allow analysts to alter factor levels within input files, rather than burying factor level settings deep inside the code. Simulation software developers should incorporate experimental design modules, particularly those involving simulation-specific factors, into their software packages. Software developers should continue developing tools that facilitate experimentation in distributed computing environments. Statistical software vendors should continue adding design, analysis, and visualization tools that address the three primary goals of simulation experiments. Simulation consultants should consider whether their clients' needs might be best served by incorporating experimental design approaches. Finally, we challenge simulation researchers and practitioners to continue a dialogue that leads to rapid dissemination of new developments in—and useful applications of—the design and analysis of simulation experiments.

Acknowledgments

We thank the Associate Editor and three referees for their very useful comments on the first version of this paper. This work was supported in part by grants from the U.S. Marine Corps Combat Development Command and the U.S. Marine Corps Warfighting Laboratory.

References

Alam, F. M., K. R. McNaught, T. J. Ringrose. 2003. A comparison of experimental designs in the development of a neural network simulation metamodel. *Simulation Modelling: Practice and Theory*. Forthcoming.

Antioniadis, A., D. T. Pham. 1998. Wavelet regression for random or irregular design. *Computational Statistics and Data Analysis* **28** 353-369.

Angün, E., D. den Hertog, G. Gürkan, J. P. C. Kleijnen. 2002. Response surface methodology revisited. E. Yücesan, C. H. Chen, J. L. Snowdon, J. M. Charnes, eds. *Proceedings of the 2002 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 377-383.

Banks, J., J. S. Carson, B. L. Nelson, D. M. Nicol. 2000. *Discrete-event Simulation*, 3rd ed. Prentice-Hall, Upper Saddle River, NJ.

Barry, P., A. Forsyth. 2003. Perspectives on distillations. G. Horne, S. Johnson eds. *Maneuver Warfare Science 2003*. USMC Project Albert, Quantico, VA. 177-193.

Batmaz, I, S. Tunali. 2003. Small response surface designs for metamodel estimation. *European Journal of Operational Research* **145** 455-470.

Bettonvil, B., J. P. C. Kleijnen. 1997. Searching for important factors in simulation models with many factors: sequential bifurcation. *European Journal of Operational Research* **96** 180-194.

Bettonvil, B., J. P. C. Kleijnen. 1990. Measurement scales and resolution IV designs. *American Journal of Mathematical and Management Sciences* **10** 309-322.

Box, G. E. P., R. Draper. 1987. *Empirical Model-building with Response Surfaces*. John Wiley & Sons, New York.

Box, G. E. P., W. G. Hunter, J. S. Hunter. 1978. *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. John Wiley & Sons, New York.

Cabrera-Rios, M., C. A. Mount-Campbell, S. A. Irani. 2002. An approach to the design of a manufacturing cell under economic considerations. *International Journal of Production Economics* **78** 223-237.

Campolongo, F., J. P. C. Kleijnen, T. Andres. 2000. Screening methods. A. Saltelli, K. Chan, E. M. Scott, eds. *Sensitivity Analysis*. John Wiley & Sons, New York. 65-89.

Cheng, R. C. H. 1997. Searching for important factors: sequential bifurcation under uncertainty. S. Andradottir, K. J. Healy, D. H. Withers, B. L. Nelson, eds. *Proceedings of the 1997 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 275-280.

Cheng, R. C. H., J. P. C. Kleijnen. 1999. Improved design of simulation experiments with highly heteroskedastic responses. *Operations Research* **47** 762-777.

Chick, S. E., K. Inoue. 2001. New procedures to select the best simulated system using common random numbers. *Management Science* **47** 1133-1149.

Cioppa, T. M. 2002. Efficient Nearly Orthogonal and Space-filling Experimental Designs for High-dimensional Complex Models. Ph.D. Dissertation, Operations Research Department, Naval Postgraduate School, Monterey, CA. http://library.nps.navy.mil/uhtbin/hyperion-image/ 02sep_Cioppa_PhD. pdf.

Cioppa, T. M., T. W. Lucas, S. M. Sanchez. 2004. Military applications of agent-based models. *Proceedings of the 2004 Winter Simulation Conference*, eds. R.G. Ingalls, M.D. Rossetti, J.S. Smith, B.A. Peters. Institute of Electrical and Electronics Engineers, Piscataway, NJ. Forthcoming.

Clarke, S. M., J. H. Griebsch, T. W. Simpson. 2003. Analysis of support vector regression for approximation of complex engineering analyses. *Proceedings of DETC '03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago*.

Cressie, N. A. C. 1993. *Statistics for Spatial Data.* Revised ed. John Wiley & Sons, New York.

Dean, A. M., S. M. Lewis. 2004. *Screening*. Springer-Verlag, New York.

Dewar, J. A., S. C. Bankes, J. S. Hodges, T. W. Lucas, D. K. Saunders-Newton, P. Vye. 1996. Credible Uses of the Distributed Interactive Simulation (DIS) system. MR-607-A, RAND, Santa Monica, CA. http://www.rand.org/publications/MR/MR607.pdf.

Donohue, J. M., E. C. Houck, R. H. Myers. 1993. Simulation designs and correlation induction for reducing second-order bias in first-order response surfaces. *Operations Research* **41** 880-902.

Fang, K. T., Y. Wang. 1994. *Number-Theoretic Methods in Statistics*. Chapman & Hall, London.

Fu, M. C. 2002. Optimization for simulation: theory vs. practice. *INFORMS J. Computing* **14** 192-215.

Gentle, J. E. 2002. *Computational Statistics*. Springer, New York.

Gill, A., D. Grieger. 2003. Comparison of agent based distillation movement algorithms. *Military Operations Research* **8**(3), 5-16.

Giunta, A. A., L. T. Watson. 1998. A comparison of approximating modeling techniques: polynomial versus interpolating models. AIAA-98-4758.

Goldsman, D., S. -H. Kim, W. S. Marshall, B. L. Nelson. 2002. Ranking and selection for steady-state simulation: procedures and perspectives. *INFORMS J. Computing* **14** 2-19.

Helton, J. C., M. G. Marietta. 2000. Special issue: The 1996 performance assessment for the waste isolation pilot plant. *Reliability Engineering and Systems Safety* **69**(1-3), 1-454.

Hodges, J. S. 1991. Six (or so) things you can do with a bad model. *Operations Research* **39** 355-365.

Holcomb, D., D. C. Montgomery, W. M. Carlyle. 2000a. Analysis of supersaturated designs. *J. Quality Technology.* Forthcoming.

Holcomb, D., D. C. Montgomery, W. M. Carlyle. 2000b. Some combinatorial aspects, construction methods, and evaluation criteria for supersaturated designs. *Quality and Reliability Engineering International.* Forthcoming.

Horne, G., M. Leonardi, eds. 2001. *Maneuver Warfare Science 2001*. Marine Corps Combat Development Command, Defense Automated Printing Service, Quantico, VA.

Horne, G., S. Johnson, eds. 2003. *Manuever Warfare Science 2003.* USMC Project Albert, Quantico, VA.

Horne, G., S. Johnson, eds. 2003. *Manuever Warfare Science 2003.* USMC Project Albert, Quantico, VA.

Hsu, J. C. 1996. *Multiple Comparisons; Theory and Methods*. Chapman & Hall, London.

Irizarry, M. de los A., M. E. Kuhl, E. K. Lada, S. Subramanian, J. R. Wilson. 2003. Analyzing transformation-based simulation metamodels. *IIE Transactions* **35** 271-283.

Jacobson, S., A. Buss, L. Schruben. 1991. Driving frequency selection for frequency domain simulation experiments. *Operations Research* **39** 917-924.

Jin, R., W. Chen, T. Simpson. 2000. Comparative studies of metamodeling techniques under multiple modeling criteria. AIAA-2000-4801.

Jin, R, W. Chen, A. Sudjianto. 2002. On sequential sampling for global metamodeling in engineering design. *Proceedings of DETC '02, ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. DETC2002/DAC-34092, Montreal, Canada. 1-10.

Johnson, M., L. Moore, D. Ylvisaker. 1990. Minimax and maximin distance designs. *J. Statistical Planning and Inference* **26** 131-148.

Kelton, W. D., R. M. Barton. Experimental design for simulation. . Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, eds. *Proceedings of the 2003 Winter Simulation Conference,* Institute of Electrical and Electronics Engineers, Piscataway, NJ. 59-65.

Khuri, A. I. 1996. Multiresponse surface methodology. S. Ghosh, C. R. Rao, eds. *Handbook of Statistics Volume 13*. Elsevier, Amsterdam.

Kleijnen, J. P. C. 2004. Invited review: An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research*. Forthcoming.

Kleijnen, J. P. C. 1998. Design for sensitivity analysis, optimization, and validation of simulation models. J. Banks, ed. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, New York. 173-223.

Kleijnen, J. P. C. 1995. Case study: statistical validation of simulation models. *European Journal of Operational Research* **87** 21-34.

Kleijnen, J. P. C. 1987. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, New York.

Kleijnen, J. P. C. 1974-1975. *Statistical Techniques in Simulation, Volumes I and II*. Marcel Dekker, Inc., New York. (Russian translation, Publishing House 'Statistics,' Moscow, 1978.)

Kleijnen, J. P. C., B. Bettonvil, F. Persson. 2004. Finding the important factors in large discrete-event simulation: sequential bifurcation and its applications. A. M. Dean and S. M. Lewis, eds. *Screening*. Springer-Verlag, New York. Forthcoming.

Kleijnen, J. P. C., B. Bettonvil, F. Persson. 2003. Robust solutions for supply chain management: Simulation and risk analysis of the Ericsson case study. Working Paper, Tilburg University, Tilburg, The Netherlands.

Kleijnen, J. P. C., R. C. H. Cheng, V. B. Melas. 2000. Optimal design of experiments with simulation models of nearly saturated queues. *Journal of Statistical Planning and Inference* **85** 19-26.

Kleijnen, J. P. C., A. J. Feelders, R. C. H. Cheng. 1998. Bootstrapping and validation of metamodels in simulation. D. J. Medeiros, E. F. Watson, J. S. Carson, M. S. Manivannan, eds. *Proceedings of the 1998 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 701-706.

Kleijnen, J. P. C., E. Gaury. 2003. Short-term robustness of production management systems: a case study. *European Journal of Operational Research* **148** (2) 452-465.

Kleijnen, J. P. C., M. T. Smits. 2003. Performance metrics in supply chain management. *Journal of the Operational Research Society* **54** (5) 507-514.

Kleijnen, J. P. C., W. C. M. Van Beers. 2004a. Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society*. Forthcoming.

Kleijnen, J. P. C., W. C. M. Van Beers. 2004b. Robustness of Kriging when interpolating in random simulation with heterogeneous variances: some experiments. *European Journal of Operational Research*. Forthcoming.

Kleijnen, J. P. C., A. J. van den Burg, R. Th. van der Ham. 1979. Generalization of simulation results: practicality of statistical methods. *European Journal of Operational Research* **3** 50-64.

Kleijnen, J. P. C., A. Vonk Noordegraaf, M. Nielen. 2001. Sensitivity analysis of censored output through polynomial, logistic and tobit models: theory and case study. B. A. Peters, J. S. Smith, D. J. Medeiros, M. W. Rohrer, eds. *Proceedings of the 2001 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 486-491.

Kleijnen, J. P. C., O. Pala. 1999. Maximizing the simulation output: a competition. *Simulation* **73** 168-173.

Kleijnen, J. P. C., R. G. Sargent. 2000. A methodology for the fitting and validation of metamodels in simulation. *European Journal of Operational Research* **120** 14-29.

Kleijnen, J. P. C., W. van Groenendaal. 1992. *Simulation: A Statistical Perspective*. John Wiley & Sons, Chichester, England.

Koehler, J. R., A. B. Owen. 1996. Computer experiments. S. Ghosh, C. R. Rao, eds. *Handbook of Statistics, Volume 13*. Elsevier, Amsterdam. 261-308.

Lauren, M. K., R. T. Stephen. 2002. Map-aware non-uniform automata—a New Zealand approach to scenario modelling. *J. Battlefield Tech.* **5**(1), 27-31.

Law, A. M., W. D. Kelton. 2000. *Simulation Modeling and Analysis.* 3rd ed. McGraw-Hill, New York.

Lin, D. K. J. 1995. Generating systematic supersaturated designs. *Technometrics* **37** 213-225.

Lophaven, S.N., H.B. Nielsen, J. Sondergaard. 2002. DACE: a Matlab Kriging toolbox, version 2.0.IMM Technical University of Denmark, Lyngby. http://www.imm.dtu.dk/~hbn/dace/

Lucas, T. W., S. M. Sanchez, L. Brown, W. Vinyard. 2002. Better designs for high-dimensional explorations of distillations. G. Horne, S. Johnson eds. *Maneuver Warfare Science 2002*. USMC Project Albert, Quantico, VA. 17-46.

Lucas, T. W., S. M. Sanchez, T. M. Cioppa, A. I. Ipekci. 2003. Generating hypotheses on fighting the global war on terrorism. G. Horne, S. Johnson eds. *Maneuver Warfare Science 2003.* USMC Project Albert, Quantico, VA. 117-137.

Lucas, T. W., S. C. Bankes, P. Vye. 1997. Improving the Analytic Contribution of Advanced Warfighting Experiments (AWEs). *RAND*, DB-207-A.

Martinez, W. L., A. R. Martinez. 2002. *Computational Statistics Handbook with MATLAB.* Chapman & Hall/CRC, Boca Raton, FL.

McKay, M. D., R. J. Beckman, W. J. Conover. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** 239-245.

Moeeni, F., S. M. Sanchez, A. J. Vakharia. 1997. A robust design methodology for Kanban system design. *International Journal of Production Research* **35** 2821-2838.

Montgomery, D. C. 2000. *Design and Analysis of Experiments.* 5th ed. John Wiley & Sons, New York.

Morrice, D. J., I. R. Bardhan. 1995. A weighted least squares approach to computer simulation factor screening. *Operations Research* **43** 792-806.

Morris, M. D., T. J. Mitchell. 1995. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* **43** 381-402.

Meyer, T., S. Johnson. 2001. Visualization for data farming: a survey of methods. G. Horne, M. Leonardi, eds. *Maneuver Warfare Science 2001*. Marine Corps Combat Development Command: Quantico, VA. 15-30.

Myers, R. H., A. I. Khuri, G. Vining. 1992. Response surface alternatives to the Taguchi robust design parameter approach. *The American Statistician* **46** 131-139.

Myers, R. H., D. C. Montgomery. 2002. *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. 2nd ed. John Wiley & Sons, New York.

Nakayama, M. Analysis of simulation output. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, eds. *Proceedings of the 2003 Winter Simulation Conference,* Institute of Electrical and Electronics Engineers, Piscataway, NJ. 49-58.

Nelson, B. L., D. Goldsman. 2001. Comparisons with a standard in simulation experiments. *Management Science* **47** 449-463.

Neter, J., W. Wasserman, M. H. Kutner. 1985. Applied Linear Statistical Models; Regression Analysis of Variance, and Experimental Design. 2nd ed. Richard D. Irwin, Inc., Homewood, IL.

Pacheco, J., C. Amon, S. Finger. 2003. Incorporating Information from Replications into Bayesian Surrogate Models. *2003 ASME Design Engineering Technical Conference* DETC2003/DTM-48644, Chicago, IL.

Pukelsheim, F. 1993. *Optimal Design of Experiments*. John Wiley & Sons, New York.

Ramberg, J. S., S. M. Sanchez, P. J. Sanchez, L. J. Hollick. 1991. Designing simulation experiments: Taguchi methods and response surface metamodels. B. L. Nelson, W. D. Kelton, G. M. Clark, eds. *Proceedings of the 1991 Winter Simulation Conference.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 167-176.

Rechtschaffner, R. L. 1967. Saturated fractions of 2n and 3n factorial designs. *Technometrics* **9** 569-575.

Sacks, J., W. J. Welch, T. J. Mitchell, H. P. Wynn. 1989. Design and analysis of computer experiments (includes Comments and Rejoinder). *Statistical Science* **4** 409-435.

Saltelli, A., S. Tarantola, P. S. Chan. 1999. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics* **41** 39-56.

Sanchez, P. J., A. H. Buss. 1987. A model for frequency domain experiments. A. Thesen, H. Grant, W. D. Kelton, eds. *Proceedings of the 1987 Winter Simulation Conference.* Institute of Electrical and Electronics Engineers, Piscataway, NJ. 424-427.

Sanchez, S. M. 2000. Robust design: seeking the best of all possible worlds. J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick, eds. *Proceedings of the 2000 Winter Simulation Conference.* Institute of Electrical and Electronics Engineers, Piscataway, NJ. 69-76.

Sanchez, S. M., L. D. Smith, E. C. Lawrence. 2001. Tolerance design revisited: assessing the impact of correlated noise factors. Working paper, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Sanchez, S. M, T. W. Lucas. 2002. Exploring the world of agent-based simulations: simple models, complex analyses. E. Yucesan, C.-H. Chen, J. L. Snowdon, J. Charnes, eds. *Proceedings of the 2002 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 116-126.

Sanchez, S. M., P. J. Sanchez, J. S. Ramberg. 1998. A simulation framework for robust system design. B. Wang ed. *Concurrent Design of Products, Manufacturing Processes and Systems*. Gordon and Breach, NY. 279-314.

Sanchez, S. M., P. J. Sanchez, J. S. Ramberg, F. Moeeni. 1996. Effective engineering design through simulation. *International Transactions on Operational Research* **3** 169-185.

Sanchez, S. M., H.-F. Wu. Frequency-based designs for terminating simulations: A peace-enforcement application. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, eds. *Proceedings of the 2003 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 952-959.

Santner, T. J., B. J. Williams, W. I. Notz. 2003. *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York.

Sasena, M. J., P. Y. Papalambros, P. Goovaerts. 2002. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization* **34** 263-278.

Schruben, L. W., V. J. Cogliano. 1987. An experimental procedure for simulation response surface model identification. *Communications of the ACM* **30** 716-730.

Schmeiser, B. W. 1982. Batch size effects in the analysis of simulation output. *Operations Research* **30** 556-568.

Shoemaker, A. C., K.-L. Tsui, C. F. J. Wu. 1991. Economical experimentation methods for robust design. *Technometrics* **33** 415-427.

Simon, H. A. 1981. *The Sciences of the Artificial.* 2nd ed. MIT Press, Cambridge, MA.

Simpson, T. W., D. K. J. Lin, W. Chen. 2001. Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications* **2**:3 209-240.

Smith, L. D., S. M. Sanchez. 2003. Assessment of business potential at retail sites: empirical findings from a U. S. supermarket chain. *The International Review of Retail, Distribution and Consumer Research* **13** 37-58.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization; Estimation, Simulation, and Control*. John Wiley & Sons, New York.

Steiger, N. M., J. R. Wilson. 2002. An improved batch means procedure for simulation output analysis. *Management Science* **48** 1569-1586.

Sugiyama, S. O., J. W. Chow. 1997. @Risk, Riskview and BestFit. *OR/MS Today* **24** 64-66.

Taguchi, G. 1987. *System of Experimental Designs, Vol. 1 and 2*. UNIPUB/Krauss International, White Plains, NY.

Trocine L., L. C. Malone. 2001. An overview of newer, advanced screening methods for the initial phase in an experimental design. B. A. Peters, J. S. Smith, D. J. Medeiros, M. W. Rohrer, eds. *Proceedings of the 2001 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 169-178.

Tunali, S., I. Batmaz. 2000. Dealing with the least squares regression assumptions in simulation metamodeling. *Computers & Industrial Engineering* **38** 307-320.

Tufte, E. R. 1990. *Envisioning Information.* Graphics Press, Cheshire, CT.

Ye, K. Q. 1998. Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association—Theory and Methods* **93** 1430-1439.

Van Beers, W.C.M., J.P.C. Kleijnen 2004. Kriging interpolation in simulation: a survey. *Proceedings of the 2004 Winter Simulation Conference*, eds. R.G. Ingalls, M.D. Rossetti, J.S. Smith, B.A. Peters. Institute of Electrical and Electronics Engineers, Piscataway, NJ. Forthcoming.

Van Beers, W.C.M., J.P.C. Kleijnen 2003. Kriging for interpolation in random simulation. *Journal of the Operational Research Society* **54** 255-262.

Vinyard, W., T. W. Lucas. 2002. Exploring combat models for non-monotonicities and remedies. *PHALANX* **35** 19, 36-38.

Vonk Noordegraaf, A., M. Nielen, J. P. C. Kleijnen. 2002. Sensitivity analysis by experimental design and metamodeling: case study on simulation in national animal disease control. *European Journal of Operational Research*. Forthcoming.

Wan, H., B. Ankenman, B. L. Nelson. 2003. Controlled sequential bifurcation: a new factor-screening method for discrete-event simulation. *Proceedings of the 2003 Winter Simulation Conference*, eds. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice. Institute of Electrical and Electronics Engineers, Piscataway, NJ. 565-573.

Wan, S. C. 2002. An exploratory analysis on the effects of human factors on combat outcomes. M. S. Thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA. http://library.nps.navy.mil/uhtbin/hyperion-image/02Mar_Wan.pdf

Watson, A. G., R. J. Barnes. 1995. Infill sampling criteria to locate extremes. *Mathematical Geology* **27** 589-608.

WebDOE http://www.webdoe.cc/

Wolf, E. S. 2003. Using agent-based distillations to explore logistics support to urban, humanitarian assistance/disaster relief operations. M.S. Thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA. http://library.nps.navy.mil/uhtbin/hyperion-image/03sep_Wolf.pdf.

Wolf, E. S., S. M. Sanchez, N. Goerger, L. Brown. 2003. Using agents to model logistics. Working Paper, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Wu, H. -F. 2002. Spectral analysis and sonification of simulation data generated in a frequency domain experiment. M. S. Thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA. http://library.nps.navy.mil/uhtbin/hyperion-image/02sep_Wu.pdf.

Zeigler B. P., K. Praehofer, T. G. Kim. 2000. *Theory of Modeling and Simulation*. 2nd ed. Academic Press, San Diego, CA.